

UNIVERSIDADE FEDERAL DO PARANÁ

MARIANE GAVIOLI BERGAMINI

**ESTIMAÇÃO DE PARÂMETROS DE SISTEMAS NÃO LINEARES UTILIZANDO
UM SISTEMA COLABORATIVO DE METAHEURÍSTICAS**

CURITIBA

2018

MARIANE GAVIOLI BERGAMINI

**ESTIMAÇÃO DE PARÂMETROS DE SISTEMAS NÃO LINEARES UTILIZANDO
UM SISTEMA COLABORATIVO DE METAHEURÍSTICAS**

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica, Áreas de Concentração Sistemas Eletrônicos, Departamento de Engenharia Elétrica, Setor de Tecnologia, da Universidade Federal do Paraná, como parte das exigências para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Gideon Villar Leandro
Coorientador: Prof. Dr. Gustavo H. da Costa Oliveira

CURITIBA
2018

Catálogo na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

- B493 Bergamini, Mariane Gavioli
Estimação de parâmetros de sistemas não lineares utilizando um sistema colaborativo de metaheurísticas / Mariane Gavioli Bergamini. – Curitiba, 2018. 93f. . : il. [algumas color.] ; 30 cm.
- Dissertação (mestrado) - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-graduação em Engenharia Elétrica, 2018.
- Orientador: Gideon Villar Leandro
Coorientador: Gustavo H. da Costa Oliveira.
Bibliografia: p. 89-93.
1. Metaheurística. 2. Sistemas de energia elétrica. I. Universidade Federal do Paraná. II. Leandro, Gideon Villar. III. Oliveira, Gustavo H. da Costa. IV. Título.

CDD: 621.39

Bibliotecária: Roseny Rivelini Morciani CRB-9/1585



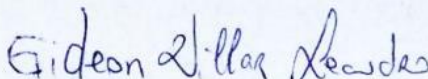
MINISTÉRIO DA EDUCAÇÃO
SETOR TECNOLOGIA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO ENGENHARIA
ELÉTRICA

TERMO DE APROVAÇÃO


Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA ELÉTRICA da Universidade Federal do Paraná foram convocados para realizar a arguição da dissertação de Mestrado de **MARIANE GAVIOLI BERGAMINI** intitulada: **ESTIMAÇÃO DE PARÂMETROS DE SISTEMAS NÃO LINEARES UTILIZANDO UM SISTEMA COLABORATIVO DE METAHEURÍSTICAS**, após terem inquirido a aluna e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

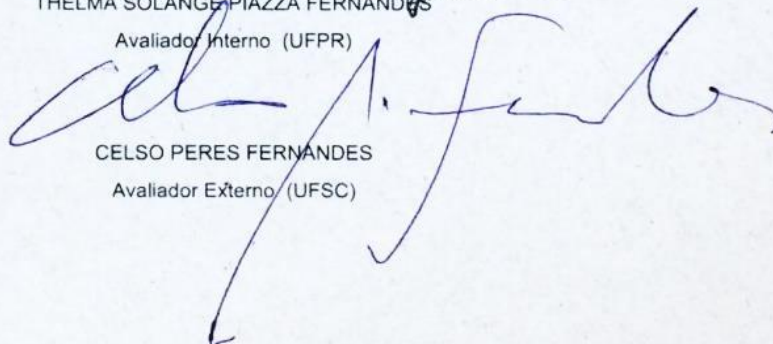
A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 23 de Fevereiro de 2018.


GIDEON VILLAR LEANDRO

Presidente da Banca Examinadora (UFPR)


THELMA SOLANGE PIAZZA FERNANDES
Avaliador Interno (UFPR)


CELSON PERES FERNANDES
Avaliador Externo (UFSC)

*Dedico este trabalho em especial à
minha família, pela confiança e força do
início ao fim desta jornada.*

AGRADECIMENTOS

Primeiramente agradeço a Deus, que me guiou o tempo todo e me manteve focada nos meus estudos para ter forças de nunca desistir de alcançar meus sonhos.

À minha mãe Maura e meu pai Edval pela compreensão, conselhos durante toda minha jornada e por não medirem esforços ao me ajudar a chegar até o exato momento e conquistar mais um sonho. À minha irmã Nayara e meu sobrinho Eduardo por se manterem presentes o tempo todo. À minha madrinha Leni por todo afeto e apoio emocional.

Ao Felipe pela paciência plena, carinho, ajuda emocional, afetiva desde o início da minha caminhada, onde se manteve presente e por toda sua compreensão.

Ao Prof. Dr. Gideon Villar Leandro pela credibilidade, paciência, aprendizado, otimismo e disponibilidade de me orientar.

Ao Prof. Dr. Gustavo Henrique da Costa Oliveira pelo aprendizado, por todas as contribuições ao trabalho e a mim.

Aos demais professores e funcionários da UFPR que contribuíram de forma direta e indireta durante a minha caminhada.

À equipe LaSico: Carla, Cristiano, Carlos e Jhonys, pela convivência, amizade e troca de conhecimento.

*“Se eu vi mais longe, foi por estar
de pé sobre ombros de gigantes.”*

ISAAC NEWTON

RESUMO

As metaheurísticas vêm sendo utilizadas cada vez mais para resolver problemas que sistemas dinâmicos estão envolvidos, pois estes requerem técnicas robustas e de fácil implementação. Com base nas metaheurísticas populacionais, vários autores apresentaram algoritmos para o problema de estimação de parâmetros não lineares e através de testes numéricos indicaram que a precisão da estimativa é satisfatória. No entanto, as taxas de convergências de algumas dessas metaheurísticas são relativamente lentas quando múltiplos parâmetros desconhecidos em um sistema dinâmico não linear são estimados simultaneamente. Como contribuição deste trabalho, três métodos colaborativos de metaheurísticas paralelizadas para estimação de parâmetros de sistemas dinâmicos não lineares são propostos. A ideia principal deste método é processar de forma paralela as metaheurísticas (Evolução Diferencial (ED), Algoritmo de Otimização de Leão (AOL) e Busca Local Iterativa (BLI)) e trocar informações entre si, para melhorar seus respectivos desempenhos quando considerados individualmente. Adicionalmente, cada algoritmo contém um processamento paralelo que visa reduzir o custo computacional. Os algoritmos propostos são comparados nas seguintes versões: clássica sem paralelo, colaborativa sem paralelo, clássica com paralelo e colaborativa com paralelo. Através dos resultados obtidos por uma análise estatística utilizando o erro médio quadrático (EMQ), as soluções dos algoritmos propostos são menores que os métodos na versão clássica. Neste mesmo cenário, a técnica de paralelismo aplicada para acelerar o cálculo da função objetivo consegue diminuir o tempo de processamento de cada algoritmo.

Palavras-chave: Metaheurísticas. Sistemas dinâmicos não lineares. Estimação de parâmetros. Técnicas de paralelismo.

ABSTRACT

The metaheuristics are used to solve problems where dynamic systems are involved, since it requires robust and easy-to-implement techniques. Based on the population metaheuristics, a lot of authors used algorithms to the nonlinear parameters estimation problem and through numerical tests indicated that the estimation precision was satisfactory. However, the convergence rates of some metaheuristics are relatively slow when multiple unknown parameters in a nonlinear dynamic system are simultaneously estimated. As a contribution of this paper, three collaborative methods of parallelized metaheuristics for nonlinear dynamic system parameter estimation are proposed. The main idea of the proposed method is to process in parallel mode the metaheuristics (Differential Evolution (DE), Lion Optimization Algorithm (LOA) and Iterative Local Search (ILS)), and exchange information between them in order to improve their own performances when considered individually. Additionally, each proposed algorithm has a parallel processing to reduce the computational cost. The proposed algorithms are compared in the following versions: unparallelized classic, unparallelized collaborative, classic with parallel and collaborative with parallel. Through the results obtained by a statistical analysis using mean square error (MSE), it shows that the solutions of the proposed algorithms are smaller than the classic version. In this scenario, the parallelism technique applied to accelerate the calculation of objective function can reduce the processing time of each algorithm.

Key-words: Metaheuristics. Nonlinear dynamics systems. Parameter estimation. Parallelism techniques.

LISTA DE FIGURAS

Figura 1 – Classificação das metaheurísticas.	17
Figura 2 – Processo da identificação de sistemas.	25
Figura 3 – Exemplos de modelagem matemática.	26
Figura 4 – Inicialização da população.	35
Figura 5 – Vetor diferença.	36
Figura 6 – Processo de mutação.	37
Figura 7 – Processo de seleção.	39
Figura 8 – Corte do cromossomo.	43
Figura 9 – Busca local por cadeia de binários.	46
Figura 10 – Perturbação dos ótimos globais.	47
Figura 11 – Metaheurística colaborativa ED com AOL.	49
Figura 12 – Metaheurística colaborativa ED com BLI.	50
Figura 13 – Metaheurística colaborativa AOL com BLI.	51
Figura 14 – Metaheurística sequencial e paralela.	53
Figura 15 – Ackley com metaheurísticas individuais.	59
Figura 16 – Ackley com metaheurísticas colaborativas.	59
Figura 17 – Rosenbrock com metaheurísticas individuais.	60
Figura 18 – Rosenbrock com metaheurísticas colaborativas	61
Figura 19 – Diagrama da turbina hidráulica.	63
Figura 20 – Modelo conjunto conduto/turbina.	63
Figura 21 – Parâmetros das metaheurísticas individuais sequencialmente.	66
Figura 22 – Parâmetros estimados por ED com AOL sequencialmente.	66
Figura 23 – Parâmetros estimados por ED com BLI sequencialmente.	67
Figura 24 – Parâmetros estimados por AOL com BLI sequencialmente.	67
Figura 25 – Saída das metaheurísticas individuais sequencialmente.	68
Figura 26 – Saída das metaheurísticas colaborativas sequencialmente.	68
Figura 27 – EMQ das metaheurísticas individuais sequencialmente.	70
Figura 28 – EMQ das metaheurísticas colaborativas sequencialmente.	70
Figura 29 – Convergência das soluções.	71
Figura 30 – EMQ das metaheurísticas individuais paralelizadas.	73
Figura 31 – EMQ das metaheurísticas colaborativas paralelizadas.	74
Figura 32 – Saída das metaheurísticas individuais paralelizadas.	74

Figura 33 – Saída das metaheurísticas colaborativas paralelizadas.....	75
Figura 34 – Diagrama de blocos da usina Hidrelétrica.....	76
Figura 35 – Diagrama de blocos do sistema hidráulico com a turbina.	76
Figura 36 – Modelo matemático da malha hidráulica.	77
Figura 37 – Variação dos parâmetros das metaheurísticas individuais.....	81
Figura 38 – Variância da técnica colaborativa AOL com BLI.....	81
Figura 39 – Variância da técnica colaborativa ED com AOL.....	82
Figura 40 – Variância da técnica colaborativa ED com BLI.....	82
Figura 41 – Saída da malha hidráulica com as metaheurísticas individuais.	82
Figura 42 – Saída da malha hidráulica com metaheurísticas colaborativas.....	83
Figura 43 – Variação da solução ótima nas iterações.....	85
Figura 44 – EMQ das metaheurísticas individuais.	85
Figura 45 – EMQ das metaheurísticas colaborativas.....	86

LISTA DE ABREVIATURAS E SIGLAS

ABGH	- Algoritmo de Busca Gravitacional Híbrido
AG	- Algoritmo Genético
AFL	- Algoritmo Formiga Leão
ALU	- Unidade Lógica e Aritmética
AM	- Algoritmo de Morcego
AOL	- Algoritmo de Otimização de Leão
ARX	- Auto Regressivo com Entradas Exógenas
ARMAX	- Auto Regressivo de Média Móvel com Entradas Exógenas
BLI	- Busca Local Iterativa
BLG	- Busca Local Guiada
BT	- Busca Tabu
CPU	- Unidade Central de Processamento
ED	- Evolução Diferencial
EE	- Estratégia de Evolução
GRASP	- Procedimento de Pesquisa Adaptativa Randomizada Gulosa
MIMD	- Fluxo Múltiplo de Instruções Múltiplas Fluxo Múltiplo de Dados
MMQ	- Método dos Mínimos Quadrados
MSE	- Erro Médio Quadrático
NARX	- Não Linear Auto Regressivo com Entradas Exógenas
NARMAX	- Não Linear Auto Regressivo de Média Móvel e Entradas Exógenas
NFIR	- Não Linear de Resposta de Impulso Finito
NOBF	- Funções Não Lineares de Bases Ortogonais
NOE	- Não Linear Erro de Saída
OCF	- Otimização de Colônia de Formigas
OE	- Erro de Saída
OLC	- Otimização de Lobo cinzento
OPE	- Otimização de Partícula de Enxames
PG	- Programação Genética
PE	- Programação Evolutiva
PID	- Controlador Proporcional Integral
RNA	- Redes Neurais Artificiais
RS	- Recozimento Simulado

SIN	- Sistema Interligado Nacional
VVS	- Vizinhaça Variável de Busca

LISTA DE SÍMBOLOS

® - marca registrada

SUMÁRIO

1.	INTRODUÇÃO	16
1.1	ESTIMAÇÃO DE PARÂMETROS COM METAHEURÍSTICAS	20
1.2	OBJETIVOS	22
1.2.1	Objetivos Específicos	22
1.3	JUSTIFICATIVA.....	23
1.4	ESTRUTURA DA DISSERTAÇÃO	24
2	ESTIMAÇÃO DE PARÂMETROS DE SISTEMAS NÃO LINEARES	25
2.1	COLETA DE DADOS.....	27
2.2	ESTRUTURA E REPRESENTAÇÃO DO MODELO.....	27
2.3	OTIMIZAÇÃO RESTRITA.....	28
2.4	ESTIMAÇÃO DE PARÂMETROS.....	29
2.5	VALIDAÇÃO DO MODELO	31
3	TÉCNICAS DE OTIMIZAÇÃO.....	32
3.1	EVOLUÇÃO DIFERENCIAL	33
3.1.1	População inicial.....	34
3.1.2	Função objetivo	36
3.1.3	Mutação.....	36
3.1.4	Cruzamento	38
3.1.5	Seleção.....	39
3.2	ALGORITMO DE OTIMIZAÇÃO DE LEÃO	40
3.2.1	População inicial.....	41
3.2.2	Função objetivo	42
3.2.3	Seleção.....	42
3.2.4	Cruzamento	43
3.2.5	Mutação.....	43
3.2.6	População nômade.....	44
3.2.7	Disputa territorial.....	44
3.2.8	Aquisição territorial	45
3.3	BUSCA LOCAL ITERATIVA	46
3.3.1	População inicial.....	47
3.3.2	Busca local	47
3.4	METAHEURÍSTICAS COLABORATIVAS	49

4	COMPUTAÇÃO PARALELA APLICADA A METAHEURÍSTICAS	52
4.1	PARALELISMO	53
4.2	TIPOS DE PARALELISMO	54
5	RESULTADOS	56
5.1	PROBLEMAS TESTES DE OTIMIZAÇÃO	56
5.2	CONJUNTO CONDUTO/TURBINA HIDRÁULICA	62
5.2.1	Metaheurísticas rodadas de forma sequencial.	64
5.2.2	Metaheurísticas rodadas de forma paralela.	71
5.3	SISTEMA DE CONTROLE DE VELOCIDADE	76
6	CONSIDERAÇÕES FINAIS	87
	REFERÊNCIAS	89

1. INTRODUÇÃO

A estimação de parâmetros é considerada uma etapa fundamental na identificação de sistemas. A identificação de sistemas é uma área da ciência que desenvolve modelos matemáticos para representar o comportamento dinâmico de sistemas reais, sendo que tais modelos correlacionam diversas variáveis observáveis ou conceituais para a descrição do comportamento do sistema sob estudo (NASH; WALKER-SMITH, 1995, MOHD AZMI *et al.*, 2015).

Estimação de parâmetros é um procedimento no qual o objetivo é minimizar uma função de erro (diferença entre uma variável temporal do sistema e do modelo), esta minimização está diretamente ligada aos valores dos parâmetros encontrados pelo método utilizado. Este procedimento de minimização geralmente é visto como um procedimento de otimização, pois os valores corretos dos parâmetros levam a um erro menor.

Existem diversos métodos de estimação de parâmetros, tanto para modelos lineares como para os modelos não lineares (COELHO; COELHO, 2004; AGUIRRE, 2007). Para modelos lineares discretos, utiliza-se o método dos mínimos quadrados, este pode ser visto como uma técnica de otimização. Já, no caso dos modelos não lineares utilizam-se os métodos metaheurísticos como técnica de otimização.

Abordagens clássicas de otimização não foram eficientes ao lidar com o problema de estimação de parâmetros de sistemas dinâmicos não lineares, devido a não linearidade dos parâmetros desses modelos. Com isto, as soluções encontradas eram ruins quando a estimação de parâmetros era realizada por mínimos quadrados (COELHO; COELHO, 2004).

Desta forma, os modelos não lineares necessitavam de técnicas mais robustas e eficientes de otimização, como são os métodos metaheurísticos, que beneficiam de operadores estocásticos, conseguindo cobrir uma região maior de possíveis soluções, e são de fácil implementação, conseguindo tratar do problema de estimação de parâmetros de modelos não-lineares (SEVERINO *et al.*, 2016).

Atualmente, as metaheurísticas ganharam ênfase por serem de fácil desenvolvimento e por encontrarem soluções factíveis mais rápido que as abordagens clássicas para problemas reais ligados às diversas áreas. As metaheurísticas não determinísticas tem a aleatoriedade (ou operadores estocásticos) como principal característica e que as diferem das metaheurísticas

determinísticas. Dessa forma as metaheurísticas não determinísticas são mais robustas, pois têm mais chances de encontrar o mínimo global mais vezes do que as metaheurísticas determinísticas (BÄCK *et al.*, 1997).

Os problemas reais de diversas áreas são considerados da classe de problemas polinomiais não determinísticos ($Np - difíceis$), pois o tempo de execução é considerado pseudopolinomial.

Desta maneira, as metaheurísticas, são métodos de otimização alternativos para resolver problemas de grandes escalas e complexos (sistemas de controle, turbina hidráulica, análise de dados estatísticos, construção de modelos, etc.) considerados $Np - difíceis$. Além disso, as metaheurísticas encontram soluções satisfatórias em tempo razoável, o que satisfaz a necessidade de agilidade dos problemas reais (RAO, 2009; TALBI, 2009).

As metaheurísticas têm sua construção baseadas na mimetização de algum processo biológico, físico, social, etc. A Figura 1 mostra algumas das metaheurísticas mais conhecidas e a sua classificação quanto a determinadas características (DRÉO *et al.*, 2007).

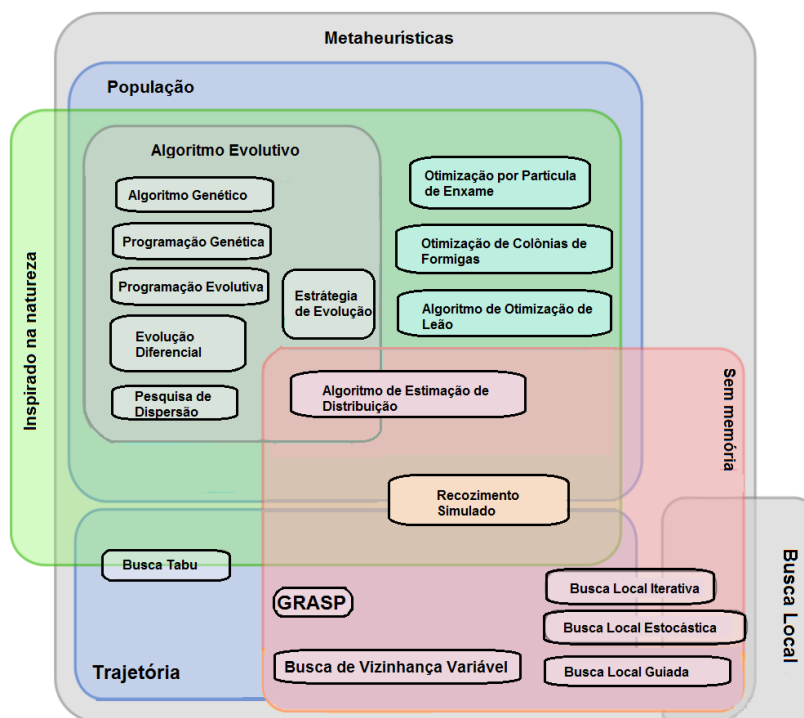


Figura 1 – Classificação das metaheurísticas.

Fonte: Adaptada de Dréo *et al* (2007).

As metaheurísticas pertencentes ao método de trajetória inicializam com uma única solução inicial, sendo esta modificada por operadores específicos que

descrevem a trajetória no espaço de busca. Essas metaheurísticas de busca focam na busca por ótimos locais ao redor de uma solução eficiente encontrada, ou seja, realiza uma busca na vizinhança da solução eficiente. Algumas dessas metaheurísticas são vistas como versões mais robustas das metaheurísticas de buscas locais clássicas. Podem-se citar algumas metaheurísticas que fazem parte dos métodos de trajetórias, como: recozimento simulado (RS) busca tabu (BT), método GRASP, vizinhança variável de busca (VVS), busca local guiada (BLG), busca local iterativa (BLI) (MELO *et al.*, 2009; ATYABI; SAMADZADEGAN, 2011; VIANNA; VIANNA, 2013).

Metaheurísticas pertencente à população de indivíduos lidam com um conjunto de soluções (indivíduos). Os métodos mais estudados estão relacionados à computação evolutiva (CE). Os métodos da CE são inspirados pela teoria evolutiva de Darwin e na teoria genética de Mendel, onde uma população de indivíduos é alterada por operadores: seleção, cruzamento (recombinação) e mutação. (BIRATTARI *et al.*, 2001; SALCEDO-SANZ, 2016).

O algoritmo genético (AG) é uns dos métodos mais conhecidos na categoria das metaheurísticas pertencente à população de indivíduos. Ele pode ser formulado de diversas maneiras de acordo com o problema (representação dos cromossomos, estratégia de seleção, tipo de operador de cruzamento e mutação) (ANITA CHRISTALINE *et al.*, 2016).

A estratégia de evolução (EE) é outro método pertencente à população de indivíduos. Este método tem como operador de mutação uma distribuição gaussiana, onde se gera um único descendente para depois ser avaliado como o descendente mais próximo do ascendente selecionado (BOUSSAÏD *et al.*, 2013).

A programação genética (PG) adota uma estratégia de busca semelhante aos AGs, mas utiliza uma representação de programa e operadores especiais. Os indivíduos da população são considerados como programas de computador que, quando executados, representam as soluções candidatas do problema. Esses programas geralmente podem ser expressos como árvores de sintaxe e não como linhas de código. Suas variáveis e constantes, são folhas da árvore, enquanto as operações aritméticas constituem os nós internos. O terminal e os conjuntos de funções formam os alfabetos dos programas a serem realizados (GENDREAU; POTVIN, 2010).

A evolução diferencial (ED) é outro tipo de algoritmo evolutivo pertencente à população de indivíduos que se destaca por sua simplicidade e eficiência para solucionar diversos tipos de problemas complexos. A mutação é o ponto estratégico deste algoritmo, pois através dele consegue-se gerar novos indivíduos com a probabilidade grande de serem selecionados como soluções boas. A ED tem demonstrado grande eficiência ao lidar com problemas de estimação de parâmetros de sistemas reais, onde consegue obter soluções factíveis e sua convergência é rápida (ÇANAKOĞLU *et al.*, 2014; DAS, 2016; BERGAMINI *et al.*, 2017).

Metaheurísticas pertencentes à população, mais conhecidas como Inteligência de enxames tem como base o comportamento social de insetos, animais, pássaros e peixes. Nesse método não se utiliza operadores cruzamento e mutação, e sim soluções potenciais, ou seja, partículas que voam pelo espaço de busca para encontrar regiões promissoras. Essas metaheurísticas são eficazes e robustas ao lidar com problemas reais complexos. Algoritmos desta classe bastante difundidos incluem a otimização de colônias de formigas, otimização por partículas de enxame e outras metaheurísticas (CLERC, 2010; MULATI *et al.*, 2013; GUI *et al.*, 2016).

O algoritmo de otimização de leão (AOL) é um novo representante da categoria Inteligência de enxames que tem obtido bastante sucesso em resoluções de problemas *Np – difíceis*. Esta metaheurística mimetiza o comportamento social dos leões, sendo que através do seu comportamento consegue-se obter boas soluções (RAJAKUMAR, 2012; YAZDANI; JOLAI, 2016).

Ainda que estes métodos sejam eficientes e robustos, quando aplicados em determinados problemas reais as metaheurísticas acabam perdendo sua agilidade durante a execução para encontrar uma solução eficiente. A perda de agilidade está ligada a quantidade de dados e variáveis que vão ser estimadas dentro da função objetivo, o que torna improvável que algum algoritmo seja executado em tempo razoável (ALBA, 2005).

Em muitos destes problemas surgem frequentemente vários critérios de desempenho (funções objetivos), em geral, conflitantes entre si. Em otimização, a escolha do método de resolução a ser utilizado depende principalmente da razão entre a qualidade da solução gerada pelo método e o tempo gasto para encontrar essa solução. Nesse nível, a maioria dos problemas é intratável, ou seja, são

problemas para os quais é improvável que se consiga desenvolver um algoritmo exato que possa ser executado em tempo razoável.

Para problemas da classe $Np - difíceis$, os modelos matemáticos resultam em equações mais complexas que afetam o tempo de processamento computacional ao realizar o cálculo da função objetivo, pois há muitas soluções a serem testadas. Para obter uma melhora neste requisito, a computação paralela vem sendo utilizada com o intuito de melhorar o tempo computacional de processamento das metaheurísticas, com esta estratégia o cálculo da função objetivo tem sido acelerado sem que haja perda de desempenho (CRAINIC; TOULOUSE, 2003).

Dessa maneira, a estimação de parâmetros na identificação de sistemas não lineares é vista como um problema que pode ser tratado de diferentes modos, um deles é através do uso de metaheurísticas.

1.1 ESTIMAÇÃO DE PARÂMETROS COM METAHEURÍSTICAS

SCHWAAB *et al.* (2008) enfatizam o uso da metaheurística otimização de partícula de enxame (OPE) ao solucionar problemas da estimação de parâmetros em modelos não lineares com eficiência ao obter resultados satisfatórios com testes realizados ao analisar a região de confiabilidade dos parâmetros estimados.

CHU e CHANG (2009) citam a eficiência da otimização de partícula de enxame (OPE) para solucionar o problema de estimação de roteamento de inundações para modelos hidrológicos onde melhora o comportamento do modelo, pois evita a estimação subjetiva dos parâmetros estimados.

GOMES *et al.* (2009) tratam do problema da estimação de parâmetros para sistemas não lineares, utilizando o modelo discreto do tipo NARX. O algoritmo genético (AG) proposto é codificado em ponto flutuante e possui modificações em relação ao algoritmo padrão tais como: normalização do espaço de busca, utilização de múltiplos operadores de cruzamento, utilização de heurística na etapa de seleção. Os resultados obtidos mostram que os parâmetros estimados para os modelos conseguem reproduzir o comportamento dinâmico não linear dos sistemas sob investigação.

NAVARRO *et al.* (2009) realizam a estimação de parâmetros de um motor de corrente contínua utilizando redes neurais artificiais (RNA) do tipo *feedforward*

multicamadas com aprendizado supervisionado pelo algoritmo de treinamento *backpropagation*. Para validar o projeto da RNA, o índice de desempenho erro médio quadrático (EMQ) foi analisado com a intenção de verificar a menor diferença entre o sistema e o modelo estimado. Esses resultados foram compilados na forma gráfica e numérica, onde é confrontado o desempenho do modelo em relação ao sistema.

LI e ZHOU (2011) estimam os parâmetros do sistema de velocidade de uma turbina hidráulica, utilizando o algoritmo de busca gravitacional, e um algoritmo híbrido de busca gravitacional com otimização de enxames de partículas. O procedimento de identificação de parâmetros é validado pela comparação de resultados experimentais e simulações. Tendo o algoritmo de busca gravitacional híbrido (ABGH) encontrado os valores dos parâmetros mais precisos do que os algoritmos de busca gravitacional, otimização de partículas de enxames e algoritmo genético.

SUN *et al.* (2012) descrevem sobre a eficiência das metaheurísticas de recozimento simulado (RS), algoritmo genético (AG), programação evolutiva (PE), evolução diferencial (ED), estratégias de evolução (EE) e algoritmos híbridos (AH) onde cada metaheurística encontra os valores dos parâmetros estimados de um sistema biológico. Esse trabalho enfatiza a importância das metaheurísticas ao solucionar os problemas da estimação de parâmetros em sistemas biológicos.

BERTACHI (2014) em seu trabalho apresenta a aplicação dos métodos de otimização de partícula de enxame e por colônia de formigas na otimização dos parâmetros de um controlador não linear de estrutura variável baseado em um controlador de variância mínima generalizada. Os resultados obtidos enfatizam a melhora do desempenho do controlador com os parâmetros otimizados.

AFSHAR-NADJAFI (2014) faz à estimação dos parâmetros da distribuição Gamma/Gompertz. Esta distribuição tem três parâmetros, e o algoritmo para a estimação dos parâmetros é baseado no método de estimativa de máxima verossimilhança. Enquanto que a busca local iterativa (BLI) é proposta para maximizar a função de verossimilhança. Finalmente, a abordagem proposta é testada computacionalmente usando alguns exemplos e resultados numéricos são analisados.

TANGHERLONI *et al.* (2016) utilizam o algoritmo de morcego (AM) e otimização de partícula de enxame (OPE) para realizar a estimação de parâmetros

de um sistema biológico. Os resultados são comparados entre as duas metaheurísticas e estas apresentam bons resultados ao se estimar os parâmetros, e no final é feito uma análise em relação ao tempo de CPU, quando essas metaheurísticas são paralelizadas, diminuindo o tempo computacional gasto dos algoritmos.

GABRIEL e SEVERINO (2016) fazem a utilização de diferentes metaheurísticas para tratar da estimação de parâmetros na identificação de sistemas não lineares. As metaheurísticas utilizadas são: algoritmo genético; otimização de enxame de partículas; e algoritmo de morcego. Para a validação dos resultados obtidos, eles usam os parâmetros estimados por essas metaheurísticas e comparam o modelo estimado por meio do método dos mínimos quadrados (MMQ). Os resultados mostram que o emprego dessas metaheurísticas como técnicas de estimação de parâmetros são eficientes, destacando que estas obtiveram modelos de melhor desempenho que o modelo estimado por meio do método dos mínimos quadrados.

DAS (2016) trata a estimação de parâmetros de um sistema de radiador espacial retangular utilizando a metaheurística evolução diferencial (ED), o modelo proposto conseguiu representar de forma satisfatória o sistema, sendo que ele atinge a temperatura necessária para que ocorra o resfriamento do radiador.

No trabalho de BOSCO *et al.* (2017) faz-se a estimação de parâmetros de um controlador PI de um motor DC, os parâmetros desse controlador são estimados com a metaheurística ED. Essa metaheurística demonstrou robustez e eficiência ao realizar a estimação dos parâmetros, conseguindo obter a saída aproximada do sistema sob estudo.

1.2 OBJETIVOS

O objetivo geral deste trabalho é utilizar metaheurísticas colaborativas em paralelo para estimar parâmetros de um conjunto conduto/turbina hidráulica do tipo Francis, assim como de um controlador de velocidade juntamente com o conjunto conduto/turbina hidráulica do tipo Kaplan.

1.2.1 Objetivos Específicos

Os objetivos específicos do trabalho podem ser divididos em:

- Analisar as metaheurísticas evolução diferencial (ED), algoritmo de otimização de leão (AOL) e busca local iterativa (BLI), para problemas da identificação de sistemas não lineares.
- Avaliar a eficiência das metaheurísticas colaborativas, em funções testes de otimização e problemas da identificação de sistemas.
- Estimar os parâmetros do conjunto conduto/turbina hidráulica e controlador de velocidade com o conjunto conduto/turbina hidráulica com as metaheurísticas populacionais: evolução diferencial (ED), algoritmo de otimização de leão (AOL) e busca local iterativa (BLI).
- Validar o modelo através de simulações.
- Comparar e avaliar o tempo computacional das metaheurísticas na versão sequencial e paralela.

1.3 JUSTIFICATIVA

Com os estudos realizados até o presente momento, percebe-se que o problema da estimação de parâmetros, tem relevância para a identificação de sistemas não lineares para se obter a representação aproximada do sistema real. Com isto, a estimação de parâmetros requer técnicas de otimização que tratam com facilidade este problema na identificação de sistemas. Neste contexto a utilização de métodos de otimização colaborativa demonstra maior robustez ao encontrar soluções factíveis, pois quando trocam informações aumentam a confiabilidade e melhoram as soluções factíveis. A principal motivação para se utilizar essa técnica é proporcionar soluções mais eficientes levando a uma maior flexibilidade ao lidar com problemas de otimização em grande escala. No trabalho de RAJAKUMAR (2014), faz-se a estimação dos números de kernel utilizando o modelo bilinear padrão dando maior precisão ao modelo, comparando a saída do sistema referenciado com a saída do modelo desenvolvido. Além disso, ele comprova que o AOL é um excelente algoritmo de otimização para lidar com grandes dimensões de parâmetros em sistemas não lineares. Já BLI não se encontra nenhum tipo de trabalho que se faz o uso da estimação de parâmetros. Dessa maneira, as duas metaheurísticas AOL e BLI são novas e com nenhum tipo de aplicação para sistemas não lineares.

1.4 ESTRUTURA DA DISSERTAÇÃO

O presente trabalho está dividido em 5 capítulos. No Capítulo 1 faz-se uma introdução contendo uma contextualização sobre modelagem matemática na identificação de sistemas e técnicas de otimização. No Capítulo 2 tem-se uma descrição da estimação de parâmetros na identificação de sistemas, assim como os tipos de técnicas encontradas para solucionar este tipo de problema existente na literatura. No Capítulo 3 são apresentados os tipos de técnicas de otimização propostos para tratar com o problema de estimação de parâmetros. No Capítulo 4 é apresentado o contexto da computação paralela nas metaheurísticas, visando melhorar o processamento das metaheurísticas. No Capítulo 5, as simulações obtidas mostram a capacidade destas metaheurísticas de lidar com a estimação de parâmetros em sistemas não lineares. Por fim, destacam-se as conclusões finais e trabalhos futuros.

2 ESTIMAÇÃO DE PARÂMETROS DE SISTEMAS NÃO LINEARES

Em várias áreas da ciência e da engenharia, a obtenção de um “bom” modelo é parte fundamental para que se possa extrair, inferir, alterar o comportamento dinâmico do sistema sob estudo.

Porém para a obtenção de tal modelo existem várias questões a serem analisadas, pois caso haja omissão, simplificações de dinâmicas relevantes, o resultado pode ficar muito aquém do esperado.

Desta forma, algumas informações devem ser levadas em conta, tais como: o modelo será linear ou não linear; os parâmetros são constantes ou variáveis; etc. Com isto, uma proposta de modelo pode ser feita definindo sua estrutura, estimando seus parâmetros e finalmente validando sua resposta (comportamento dinâmico).

A área da modelagem, com foco em sistemas físicos, que utiliza dados históricos para a construção de um modelo matemático que represente as dinâmicas do sistema sob estudo é denominada identificação de sistemas.

Segundo Aguirre *et al.* (2007) e Ljung (2010), a identificação de sistemas pode ser representada pelas etapas conforme a Figura 2.

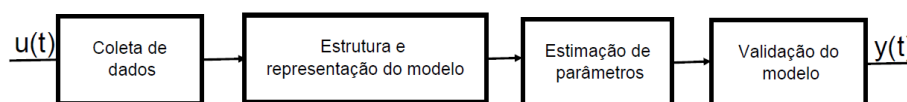


Figura 2 – Processo da identificação de sistemas.
Fonte: Adaptada de Aguirre *et al.* (2007).

Aguirre (2007) cita alguns tipos de modelagens pertencentes à identificação de sistemas, dependendo das informações conhecidas *a priori* ou não, como exemplo pode se citar: modelagem caixa branca, modelagem caixa preta e modelagem caixa cinza.

A modelagem caixa branca conhecida também por modelagem física, se refere ao conhecimento pleno do sistema, de seus parâmetros e da sua estrutura. Esse tipo de modelagem nem sempre é trivial devido ao seu alto nível de complexidade e alto dispêndio de tempo para se modelar o sistema.

A modelagem caixa preta possibilita prever e simular o comportamento dinâmico do sistema a partir de dados, medições ou observações. O comportamento do sistema é definido por variáveis internas que se comunicam produzindo sinais

observados, referentes à saída do sistema (AGUIRRE *et al.*, 2007). Além disto, não se tem nenhum conhecimento ou informação sobre o sistema a ser modelado. Os dados de entrada e saída do modelo são essenciais, pois auxiliam ao solucionar o problema do sistema, independentemente do problema notório.

A modelagem caixa cinza parte-se do princípio que se tem alguma informação *a priori* do sistema, que ajuda a construção do modelo. A partir destas informações é possível prever e simular o comportamento dinâmico do sistema a partir de dados, medições e da estrutura do modelo (LJUNG, 1987; CORRÊA; AGUIRRE, 2004).

A modelagem caixa preta é a mais utilizada nas engenharias, pois na maioria das vezes têm-se somente os dados de entradas e saídas observados do sistema, contudo seus parâmetros, estrutura de modelo não são conhecidos. Para compreender melhor a teoria da modelagem, a Figura 3 mostra algumas informações sobre a modelagem na identificação de sistemas.

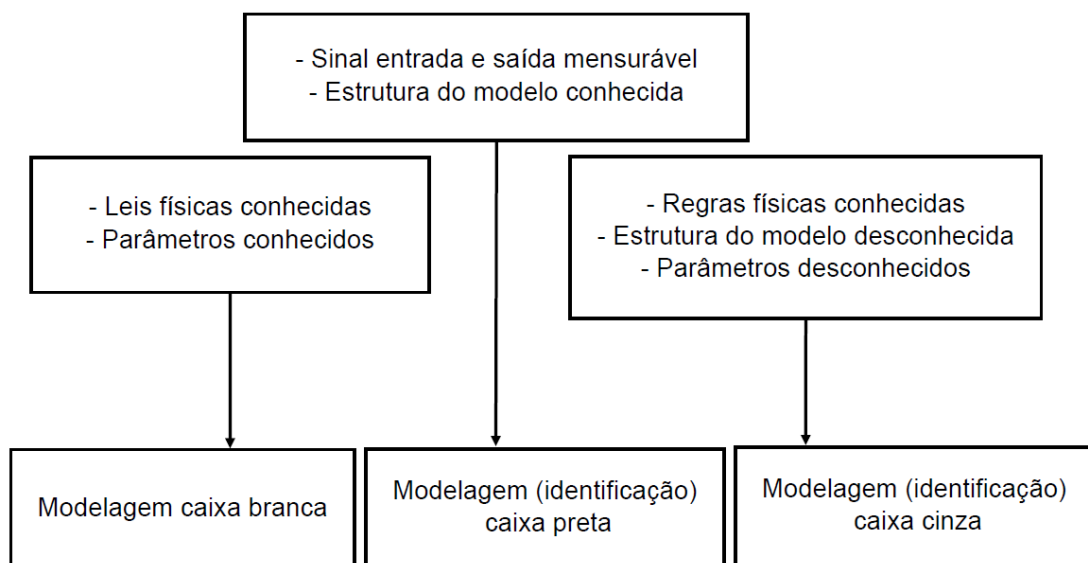


Figura 3 – Exemplos de modelagem matemática.

Fonte: Adaptada de Ljung (1987).

Nos modelos do tipo caixa preta, seus parâmetros são considerados constantes matemáticas e não possuem interpretações físicas. Mas nos modelos caixa cinza os parâmetros podem ter representações físicas (capacitância, resistência elétrica e etc). O modelo caixa preta tem conhecimento dos dados de entrada e saída do sistema que são fundamentais para reproduzir seus modelos

matemáticos. Da mesma forma, o modelo caixa cinza também tem esse conhecimento dos dados, por outro lado a estrutura do modelo é conhecida e alguns parâmetros ou variáveis são estimadas para consolidar o modelo (ISERMANN; MÜNCHHOF, 2011).

Nas seções a seguir são apresentadas as etapas da identificação de sistemas focada em sistemas dinâmicos.

2.1 COLETA DE DADOS

A primeira etapa da identificação de sistemas é a coleta de dados predeterminados, sendo apresentado pelo primeiro bloco da Figura 3. Esses dados são coletados/medidos simultaneamente onde $y(t)$ são os dados referentes à saída do sistema dinâmico e $u(t)$ são os dados referentes à entrada do sistema. Ao analisar os dados observa-se há distúrbios de frequência neles, caso afirmativo estes deve ser submetido a uma filtragem para eliminar os ruídos contidos nos sinais e assim melhorar a relação sinal ruído dos sinais dos dados de entrada e de saída.

2.2 ESTRUTURA E REPRESENTAÇÃO DO MODELO

Outro aspecto importante é a definição da estrutura do modelo. Esse aspecto esta relacionado com o tipo de problema se ele é estático ou dinâmico, linear ou não linear, a dimensão do modelo em relação à quantidade de entradas/saídas, a qualidade e quantidade de dados observados, e a complexidade do sistema (AGUIRRE *et al.*, 2007; LJUNG, 2010).

Um sistema é dito linear se ele satisfaz o princípio da superposição (princípio da aditividade e princípio da homogeneidade). Um sistema é dito não linear se ele não satisfaz o princípio da superposição. Ao classificar um modelo como não linear é necessário realizar alguns tipos de testes (tipo de não linearidade, por exemplo) e fazer comparações em relação ao comportamento do sistema, ajudando-o a encontrar soluções para tais problemas na identificação de sistemas (BORTOFF; SPONG, 1990; NELLES, 2001; LJUNG, 2010). Diante disso, os modelos

não lineares são mais complexos e demandam técnicas mais robustas para lidar melhor com o problema exposto (AGUIRRE, 2007).

Aguirre (2007) descreve algumas representações de modelos lineares, dentre eles estão: auto regressivo com entradas exógenas (ARX); e auto regressivo de média móvel com entradas exógenas (ARMAX). Além deles encontram-se o erro de saída (OE) e Box-Jenkins que são do tipo erro na saída, o modelo Box-Jenkins se baseia nos dados presentes e dados passados para prever valores futuros de uma série por intermédio da correlação temporal dos valores analisados.

Os correspondentes modelos não lineares são representados pelas estruturas, não linear auto regressivo com entradas exógenas (NARX) e não linear auto regressivo de média móvel com entradas exógenas (NARMAX). A vantagem em relação aos auto regressores dos modelos lineares encontra-se na capacidade de representar o modelo a partir da polarização do erro de estimação, e a desvantagem é o número maior de parâmetros que resulta em uma variância maior na estimação de parâmetros (CORRÊA *et al.*, 2002). A estrutura não linear do erro de saída (NOE) pode proporcionar resultados superiores em simulação com relação à estrutura NARX (PEARSON; POTTSMANN, 2000; AGUIRRE *et al.*, 2007; BARROSO *et al.*, 2007).

2.3 OTIMIZAÇÃO RESTRITA

Com a estrutura do modelo já definida, o próximo passo é a estimação de parâmetros. Como consequência é preciso escolher um algoritmo ou técnica de otimização para dar início ao processo de estimação. Vários algoritmos demonstraram serem eficientes ao estimar parâmetros de modelos não lineares devido à robustez e fácil implementação (AGUIRRE, 2007). A principal ideia da otimização não linear com restrição é minimizar ou maximizar um conjunto de parâmetros do sistema identificado, com o objetivo de melhorar os resultados. Essa minimização ou maximização ocorre através da função objetivo, a qual é formulada por equações do modelo desenvolvido que permite avaliar cada indivíduo da população. Além disso, a otimização não linear tem na variação dos parâmetros estimados, a possibilidade de pesquisar melhor por soluções candidatas factíveis, desta forma quanto menor for a variação dos parâmetros a cada processo de

estimação, significa que o algoritmo encontrou boas soluções. A otimização não linear com restrição pode ser escrita como

$$\min. f(x) = [x_1, x_2, \dots, x_i]; \quad (2.1)$$

$$\theta_i = B_N + \text{rand}(NP, 1) * (B_M - B_N) > 0. \quad (2.2)$$

onde:

$f(x)$ é a função objetivo, assumindo os valores reais de θ ,

M e N são os conjuntos de índices das restrições de igualdade e desigualdade,

θ representa as variáveis a ser estimadas.

A otimização com restrição ajuda a encontrar a solução viável da função uma vez que as restrições determinam os limites para expressar o comportamento do sistema não linear. Os problemas de otimização podem ser desenvolvidos de duas maneiras: monobjetivo e multiobjectivo. Para o presente trabalho, é utilizado monobjetivo, ou seja, o problema de otimização trabalha com uma única função objetivo a ser minimizada.

Porém, existe certa dificuldade para encontrar as soluções ótimas (soluções factíveis), pois elas dependem dos mínimos e máximos locais da função objetivo. Neste caso, a escolha do algoritmo ajuda ao encontrar por melhores soluções. Os algoritmos são divididos em duas categorias: determinísticos e não determinísticos. Os algoritmos não determinísticos apresentam grande eficiência devido à sua principal característica, aleatoriedade, que faz aumentar as chances de se alcançar soluções factíveis (STORN; PRICE, 1997; BOUSSAÏD *et al.*, 2013; BOSCO *et al.*, 2017).

2.4 ESTIMAÇÃO DE PARÂMETROS

Após a escolha da estrutura do modelo e do método de otimização, o passo a seguir é a estimação de parâmetros. A estimação de parâmetros se destaca como um dos passos mais relevantes para a identificação de sistemas. Ela juntamente com a estrutura do modelo consegue propiciar o comportamento aproximado do sistema real sob estudo (CORRÊA; AGUIRRE, 2004; MARGOTI *et al.*, 2010).

O objetivo da estimação de parâmetros é encontrar valores reais para cada parâmetro ou variável estimada, tentando fazer que a saída do modelo tenha uma quase sobreposição da saída do sistema. O modelo representado pela equação (2.1) realiza o processo da estimação de parâmetro utilizando os dados de entrada e saída do sistema (FREITAS; MACAU, 2004; GOMES *et al.*, 2009).

$$\hat{y}(N) = f(Z(N)) + g(Z(N))\theta. \quad (2.1)$$

onde:

$Z(N)$ é a matriz de dados compostas pelas entradas (u) e pelas saídas (y) do modelo,

$N=1,2,\dots, n_{amostra}$,

f e g são campos vetoriais $\in \mathbb{R}^N$,

θ é o conjunto de parâmetros a ser estimados através da técnica de otimização.

Os dados podem ser no tempo ou na frequência, segundo o modelo escolhido. Um sinal é dito de tempo contínuo, quando ele é definido para qualquer instante de tempo. Já de tempo discreto, quando ele é definido somente para instantes de tempo pré-definidos. Uma vez estimado os parâmetros, só podem ser ditos válidos quando demonstram boa similaridade da saída obtida com a saída referenciada do modelo sob estudo.

A estimação de parâmetros é realizada então após a escolha da técnica de otimização a ser utilizada, pois para modelos não lineares suas soluções são obtidas por processo iterativo numérico o qual requer a minimizar da função objetivo que irá avaliar os parâmetros estimados (PRATA *et al.*, 2009).

Aguirre (2007) e Barroso *et al.* (2007) observam que as metaheurísticas são métodos eficientes para tratar do problema de estimação de parâmetros de modelos não lineares.

Hibbert (1993) faz algumas considerações que são importantes para avaliar os parâmetros estimados, como: (a) espaço de busca dos parâmetros; (b) os mínimos locais existentes; (c) a continuidade da função objetivo; (d) a precisão ao encontrar os melhores valores para os parâmetros. As relações (c) e (d) estão ligadas a complexidade que as não linearidades trás consigo onde muitos métodos

clássicos não consegue tratar o problema de estimação de parâmetros. Já as relações (a) e (b) estão relacionadas diretamente aos valores iniciais dos parâmetros.

2.5 VALIDAÇÃO DO MODELO

A última etapa é a validação do modelo, onde a qualidade do modelo é analisada e ele pode ser considerado aceito ou rejeitado. Essa etapa é considerada essencial para a finalização do processo da identificação de sistemas, pois através dela verifica-se que a estrutura do modelo e a estimação de parâmetros conseguem validar ou não o modelo (RACHAD et al., 2015).

Segundo Aguirre *et al.* (2007) um bom modelo só é validado quando reproduz as características aproximadas ou exatas do sistema identificado. Os testes dinâmicos validam o modelo com os parâmetros estimados, utilizando os dados obtidos. Nestes testes é realizada uma análise visual comparativa com os dados do sistema identificado, determinando se o modelo está ou não adequado. A equação (2.3) ilustra o procedimento de validação dos dados (AGUIRRE *et al.*, 2007).

$$G(k) = y(k) - \hat{y}(k). \quad (2.3)$$

A equação (2.3) fornece o erro entre a saída real e a saída do sistema, porém é mais usual a utilização de métricas estatísticas para a validação do modelo. A equação (2.4), mostra uma dessas métricas, referenciada como o erro médio quadrático:

$$EMQ = \sum_{i=1}^{iter} (|y_i^{Ref} - y_i^{est}|)^2. \quad (2.4)$$

onde:

$y_i^{Ref} = y(k)$ é a saída do sistema real,

$y_i^{est} = \hat{y}(k)$ é a saída estimada,

Visto que o EMQ é uma medida de desempenho, este trabalho faz o uso dele para validar o modelo identificado.

3 TÉCNICAS DE OTIMIZAÇÃO

Alguns problemas de otimização podem ser definidos como problemas sem soluções por técnicas exatas, ou seja, técnicas determinísticas. Geralmente, estes problemas são fracionados em várias categorias (contínua ou discreta, mono ou multiobjectivo, estática ou dinâmica, por exemplo) onde se tem o único objetivo, encontrar soluções que sejam factíveis (BOUSSAÏD *et al.*, 2013).

Conforme Arroyo (2002) a escolha do método de resolução, depende da razão entre a qualidade da solução gerada por tal método e o tempo que ela gasta para encontrar a solução, na maioria das vezes são problemas intratáveis, devido ao seu tempo de processamento realizado pelo algoritmo.

A computação evolutiva e algumas heurísticas de busca têm sua inspiração na natureza. Estes métodos estão sendo usados com frequência em problemas de otimização, consideradas ferramentas importantes quando se trata de problemas *Np – difíceis* que precisam ser resolvidos (PANT *et al.*, 2009).

Mirajalili (2015) descreve os algoritmos metaheurísticos como técnicas primárias para obter soluções ótimas de problemas reais de otimização de nosso cotidiano. Estes algoritmos se beneficiam principalmente dos operadores estocásticos existentes neles, pois tem abordagens não determinísticas. Sendo, a aleatoriedade a principal característica desses algoritmos. Significa que eles operam de modo aleatório quando procuram a solução ótima global nos espaços de busca. A aleatoriedade torna o algoritmo mais confiável na obtenção da solução a cada execução, por ser um algoritmo que utiliza a formulação do problema para avaliar o conjunto de soluções, o processo é realizado independentemente do problema e com base nas entradas dadas e saídas recebidas. Conceitos evolutivos naturais ou comportamentos coletivos são inspirações para boa parte dos algoritmos deste campo. Além disso, seguem uma estrutura geral e comum, na qual um conjunto de soluções é gerado aleatoriamente aprimorando ou evoluindo iterativamente.

Nos últimos anos tem sido propostos e apresentados uma quantidade razoavelmente grande de metaheurísticas baseadas em comportamentos de indivíduos (populações) ou enxames (sociedades). Cada uma com suas vantagens e desvantagens, sendo quase impossível dizer que existe uma melhor entre todas, para todas as aplicações.

Ao longo dos anos as metaheurísticas vêm sendo aprimoradas devido aos novos problemas de alta complexidade expostos nas engenharias. As metaheurísticas têm encontrado soluções de alta qualidade para um número significativamente grande de problemas complexos que não conseguem serem resolvidos por outros métodos (BÄCK *et al.*, 1997).

Uma vantagem ao utilizar este tipo de algoritmos é devida aos valores obtidos ser mais próximos do esperado e a eficiência de convergência do ótimo global e/ou local. Além de possuir várias classes de algoritmos robustos com base na evolução genética, comportamento social, busca de vizinhança e entre outras classes (BÄCK *et al.*, 1997; EIBEN, A.E. ; SMITH, 2004; BOUSSAÏD *et al.*, 2013).

Algumas metaheurísticas que podem ser encontrada na literatura e de grande relevância para aplicações em problemas de otimização baseadas em população são: AG, ED e etc. Na categoria de metaheurísticas com base no comportamento social tem-se: AOL, ALC, OCF e etc. Já na categoria das metaheurísticas de busca na vizinhança encontram se os algoritmos: BLI, OPE, BT e etc.

Neste trabalho serão utilizadas três metaheurísticas dentre esta vasta gama disponibilizada na literatura. A seguir será apresentada em mais detalhes a evolução diferencial (ED) metaheurística baseada em população, o algoritmo de otimização de leão (AOL) metaheurística que baseia no comportamento social dos leões e por último, a técnica denominada algoritmo de busca local iterativa (BLI) que trabalha com um único caminho para chegar aos ótimos locais sem alocar o caminho percorrido, ou seja, o caminho não importa, mas sim conseguir os pontos mais próximos da vizinhança.

3.1 EVOLUÇÃO DIFERENCIAL

Em 1995, a evolução diferencial (ED) teve sua primeira publicação em forma de relatório técnico. Desde então começou a ter uma ampla variedade de aplicações no mundo real (JOUNI A. LAMPINEN, KENNETH V. PRICE, 2008).

De acordo com Kushida *et al.* (2015) ED é considerada uma metaheurística baseada na população para resolver problemas de otimização no tempo contínuo. Por causa de sua simplicidade, eficácia e robustez tornaram-se gradualmente

conhecida e é aplicada em diversos campos. Seu desempenho está diretamente ligado a sua estratégia de geração do vetor experimental, o qual depende de dois fatores principais, os operadores de mutação (F) e taxa de cruzamento (Cr). Características funcionais tais como, dimensão do problema (dim), número de mínimos locais e grau de dependência dos parâmetros dão valores adequados a estes fatores e estratégia de geração do vetor de ensaio. Muitas análises teóricas e empíricas foram realizadas sobre os parâmetros de controle. Os respectivos valores de F e Cr são definidos por valores fixos em um determinado intervalo $[0,5 \ 1]$ para F e para Cr no intervalo $[0,7 \ 1]$ ou então definidos por valores médios através de estudos existentes que testaram esses valores, e levando em conta a diversidade de convergência utilizando informações de classificação dos indivíduos.

O pseudocódigo do algoritmo ED é apresentado a seguir.

Pseudocódigo 1: Evolução Diferencial

```

Gerar população aleatoriamente
Avaliar a aptidão de cada indivíduo da população
para critério de parada ← melhor indivíduo faça
  Mutação
  Avaliar a aptidão de cada indivíduo da nova população
  Cruzamento
  Seleção
  Avaliar a aptidão da nova população
se critério de parada é válido
  fim
se não voltar ao início

```

A seguir será explanada cada linha do pseudocódigo do algoritmo ED apresentado.

3.1.1 População inicial

A população vetorial inicial ($x_{i,j}$) é composta por Np indivíduos, é escolhida aleatoriamente, e deve cobrir o espaço de busca que compõem a solução.

A população ($x_{i,j}$) é iniciada após definir os limites superior ($Bu_{i,j}$) e inferior ($Bl_{i,j}$) para cada indivíduo da população.

O vetor inicial dos indivíduos X_i é representado pela equação (3.1):

$$x_{i,j} = rand(N, dim)(Bu_{i,j} - Bl_{i,j}) + Bl_{i,j}. \quad (3.1)$$

onde:

N = tamanho da população

dim = dimensão do problema (número de variáveis estimadas)

$i = 1, 2, \dots, N_p,$

$j = 1, 2, \dots, D,$

$x_{i,j}$ é um indivíduo $\in X_i$.

Observando-se a Figura 4, a população de vetores é gerada dentro do espaço de busca (mín., máx.) e os indivíduos gerados aleatoriamente que estão perto do menor círculo correspondem aos ótimos globais e quanto mais longe estiverem do círculo menor, são ditos ótimos locais.

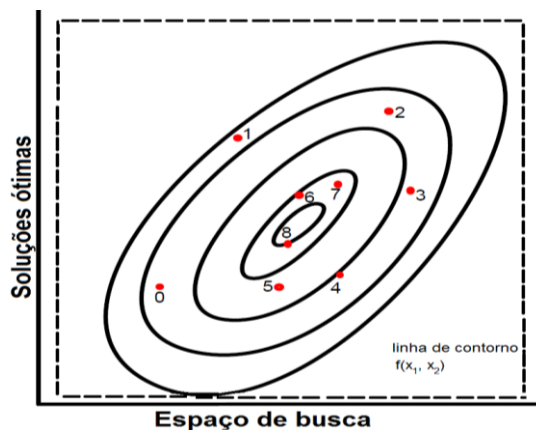


Figura 4 – Inicialização da população.
Fonte: Adaptada de Storn e Price (1997).

Como se pode visualizar na Figura 4, a região de parâmetros é inteiramente coberta e representada pela linha de contorno $f(x_1, x_2)$. Assim, os vetores obtêm um índice de contabilidade para cada um deles ao entrarem na competição dentro do espaço de busca.

3.1.2 Função objetivo

Os indivíduos da população passam por uma avaliação de aptidão referenciada como função objetivo, função de custo ou função de aptidão (fitness). A função objetivo trata da minimização ou da maximização do problema estudado, que pode ser representada pela seguinte formulação:

$$\min. f(x_i) = f(x_1, x_2, \dots, x_N). \quad (3.2)$$

A equação (3.2) representa a função de custo, pois no presente trabalho se trata do erro de predição na função, onde é utilizada a minimização. Os valores dos indivíduos avaliados são x_1, x_2, \dots, x_N , a função de custo é representada por f e o tamanho da população por N .

3.1.3 Mutaç o

No caso de solu  es preliminares, a popula  o inicial pode ser gerada por adi  o de desvios aleat rios normalmente distribu dos para uma solu  o nominal, x_{nom} . Assim, a ED gera novos vetores adicionando a diferen a ponderada entre dois vetores populacionais a um terceiro vetor. Chamamos esta opera  o de muta  o (F). Dessa maneira, s o selecionados aleatoriamente dois vetores diferen a (x_{r1} e x_{r2}) pertencentes ao vetor da popula  o. A Figura 5 ilustra os dois vetores e o vetor diferen a dentro do espa o de busca da popula  o.

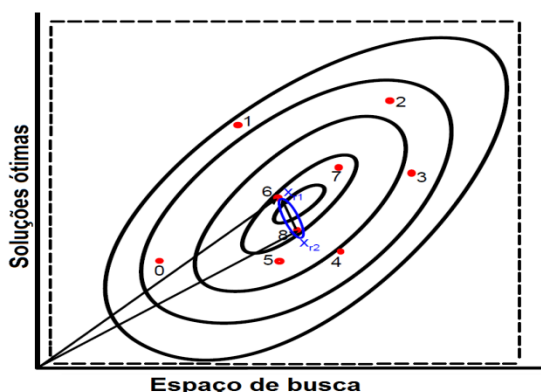


Figura 5 – Vetor diferen a.
Fonte: Adaptada de Storn e Price (1997).

Os vetores diferença (x_{r1} e x_{r2}) são sinalizados pelas retas da Figura 5. Dessa maneira, ED sofre mutação e depois sofre recombinação para produzir uma nova população de vetores, ou seja, novos indivíduos são gerados.

A mutação diferencial adiciona a diferença escalar ($x_{r1} - x_{r2}$), a um terceiro vetor x_{r3} aleatoriamente selecionado da população.

$$x_{r3} = rand(N). \quad (3.3)$$

O vetor x_{r3} ilustrado na Figura 6 é selecionado conforme Eq. (3.3) e juntamente com o vetor de diferença ponderada produz o vetor experimental u_0 .

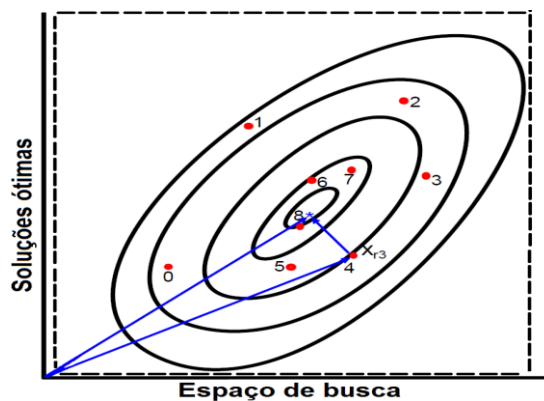


Figura 6 – Processo de mutação.
Fonte: Adaptada de Storn e Price (1997).

Com isso, o processo de mutação é representado pela equação (3.4) onde a diferença dos vetores x_{r1} e x_{r2} vezes a taxa de mutação e soma com o vetor x_{r3} .

$$u_0 = x_{r3} + F(x_{r1} - x_{r2}), \quad (3.4)$$

Devido a várias pesquisas realizadas, Zhang *et al.* (2015) apresentam um conjunto de múltiplas estratégias no operador mutação que são muitos promissores, seu desempenho pode ser melhorado, através da combinação de várias estratégias de geração de vetor de teste com alguns parâmetros de controle adequados.

Algumas estratégias que podem definir o processo de mutação são descritas a seguir (AO; CHI, 2009; KUSHIDA *et al.*, 2015).

- ED/rand/1

$$V_k = x_m + F(x_i - x_j), \quad (3.5)$$

- ED/rand/2

$$V_k = x_m + F(x_i - x_j) + F(x_k - x_l), \quad (3.6)$$

- ED/best/1

$$V_k = x_{best} + F(x_i - x_j), \quad (3.7)$$

- ED/best/2

$$V_k = x_{best} + F(x_i - x_j) + F(x_k - x_l), \quad (3.8)$$

- ED/current to best/1

$$V_k = x_m + F(x_{best} - x_i) + F(x_m - x_j). \quad (3.9)$$

onde:

$x_m, x_i, x_j, x_k, x_l \in X_i$ são obtidos de forma aleatória,

x_{best} é o valor de aptidão do melhor indivíduo encontrado,

F é a taxa de mutação que será aplicada na população,

V_k vetor mutante ou vetor mutado.

3.1.4 Cruzamento

Após o processo de mutação, ocorre o processo de cruzamento, chamado também de recombinação discreta. Os parâmetros do vetor mutado são misturados com os parâmetros de outro vetor pré-determinado, vetor alvo, responsável por produzir o vetor experimental. Esta mistura é referida com cruzamento (Cr). Sua probabilidade, $Cr \in [0 \ 1]$, é um valor definido pelo usuário que controla os valores dos parâmetros dos vetores mutantes copiados. Para saber qual fonte ele contribui com um dado parâmetro, um número é gerado de forma aleatória uniforme, $randj(0,1)$. Se o número achado for menor ou igual a Cr , então o parâmetro experimental é herdado x_{ij} . O vetor atualizado u_{ij} , é definido como:

$$u_0 = \begin{cases} v_{i,j} & \text{se } (rand(0,1) \leq Cr) \\ x_{i,j} & \text{caso contrário} \end{cases} \quad (3.10)$$

onde $rand(0,1)$ número gerado aleatoriamente dentro do intervalo $[0 \ 1]$ e Cr é a taxa de cruzamento.

3.1.5 Seleção

O operador de seleção faz a comparação do vetor teste (u_0) com o vetor alvo ($x_{i,j}$) para gerar os novos indivíduos. Se o valor de aptidão do vetor alvo for menor que o valor do vetor teste, o vetor alvo avança para a próxima geração, caso contrário, o $x_{i,j}$ é substituído pelo valor de u_0 (JOUNI A. LAMPINEN, KENNETH V. PRICE, 2008). Este processo pode ser escrito da seguinte maneira:

$$x_{i,j+1} = \begin{cases} u_0 & \text{if } f(u_0) \leq f(x_{i,j}) \\ x_{i,j} & \text{caso contrário} \end{cases} \quad (3.11)$$

onde:

$$i = 1, 2, \dots, N,$$

$$j = 1, 2, \dots, \text{dim},$$

$f(u_{i,j})$ é o valor de aptidão do melhor indivíduo do vetor teste,

$f(x_{i,j})$ é o valor de aptidão do melhor indivíduo do vetor alvo.

Para o processo de seleção, a Figura 7 ilustra este processo. O melhor indivíduo é selecionado a partir do seu valor de aptidão avaliado pela função objetivo. Neste caso, a função objetivo utiliza a minimização, com isto o melhor indivíduo será o que apresentar menor valor de aptidão.

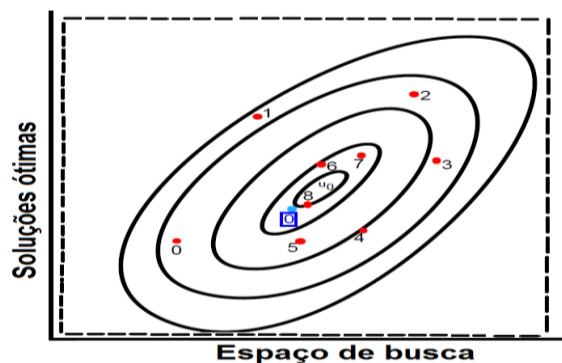


Figura 7 – Processo de seleção.
Fonte: Adaptada de Storn e Price (1997).

Ao comparar $u_{i,j}$ com $x_{i,j}$, ED integra mais fortemente a recombinação e a seleção do que outras técnicas de otimização pois as características do vetor $u_{i,j}$ são herdadas para $x_{i,j}$.

3.2 ALGORITMO DE OTIMIZAÇÃO DE LEÃO

Rajakumar (2012) e Wang *et al.* (2012) foram os primeiros a descrever sobre o algoritmo de leão onde o cita como uma nova técnica de otimização inspirado na natureza e robusto ao tratar de problemas *Np – difíceis*. Este tipo de técnica tem por base o comportamento social dos leões e é formulado para encontrar as soluções ótimas em um espaço de busca considerado grande. Uma diferença encontrada neste algoritmo em relação aos trabalhos públicos por Rajakumar (2012) é por utilizar o processo de caça no algoritmo. Já em Wang *et al.* (2012) utiliza a disputa territorial para poder encontrar os ótimos, mas nenhuns destes trabalhos estão sendo aplicados em problemas reais do nosso cotidiano e sim apenas em funções de teste de otimização como validação do potencial que este algoritmo possui ao solucionar os ótimos dentro do espaço de busca considerado grande para outras metaheurísticas que não são capazes de trabalhar com um espaço de busca significativamente grande (RAJAKUMAR, 2012; WANG *et al.*, 2012).

Quando se trata de algoritmos que tem base no comportamento social de bando ou enxames é preciso pensar em algumas características importantes, como o comportamento cooperativo (os leões) e a divisão de trabalho (as formigas). Os leões são felinos selvagens que mais apresentam cooperativismo e força dentro do bando, divididos em dois grupos: leão orgulho e leão nômade. O leão nômade é aquele que foi expulso do seu território pelo leão orgulho (leão rei) ao atingir a idade adulta (quatro anos). O leão nômade só volta ao antigo território para fazer uma disputa territorial, onde a força desses leões são medidas e o mais forte ganha à disputa. Aquele que perder é expulso definitivamente e se torna o leão nômade. Mas caso o leão nômade ganhe a disputa, ele mata os filhotes do antigo leão orgulho e a leoa escolhe ficar no bando ou ir embora com o leão expulso, isto é, no caso do leão orgulho perder a disputa territorial (MOSSER, 2008).

Mas antes de descrever todos os processos que compõem está metaheurística, na literatura pode se encontrar várias maneiras de desenvolvê-lo onde as soluções ótimas são atualizadas por meio do processo de caça (WANG *et al.*, 2012; YAZDANI; JOLAI, 2016) ou disputa territorial (RAJAKUMAR, 2012). No caso deste trabalho, foi escolhido o processo de disputa territorial para encontrar as soluções ótimas e os processos realizados.

As etapas desta metaheurística podem ser visualizadas no seguinte pseudocódigo

Pseudocódigo 2: Otimização de Leão

Gerar população aleatoriamente

Separar população entre leões e leoas

Avaliar aptidão de cada indivíduo da população

para critério de parada ← melhor indivíduo faça

Seleção

Acasalamento:

Gerar população nômade

Cruzamento

Mutação

Avaliar aptidão de cada indivíduo da nova população

Seleciona os melhores indivíduos nômades e leões orgulho

Disputa de território

Aquisição Territorial

se critério de parada for aceito

fim

se não voltar ao início

3.2.1 População inicial

A população ($x_{i,j}$) é gerada de forma aleatória através dos limites superior (B_u) e inferior (B_l) como demonstrado na equação (3.7). Sendo necessário separar a população inicial entre leão e leoa. No grupo destes felinos a maior porcentagem é determinada pelas leoas com 70% a 90% da população inicial e os leões com 30% a 10%. Pode-se demonstrar (YAZDANI; JOLAI, 2016) que

$$Leoa_{i,j} = x_{i,j} \geq r = 15\% ; \quad (3.12)$$

$$Leao_{i,j} = x_{i,j} < r = 15\% . \quad (3.13)$$

As equações (3.12) e (3.13) passam por uma avaliação através da função objetivo, o qual refere se a minimização da função objetivo.

3.2.2 Função objetivo

A função objetivo é realizada a partir da equação (3.2), onde é minimizada a função para encontrar soluções factíveis.

3.2.3 Seleção

O processo de seleção determina se o indivíduo da população é apto ou não através do método de Roleta (LIPOWSKI; LIPOWSKA, 2012). Este tipo de método é organizado por uma jogada de roleta, que geralmente processa a quantidade de vezes da população inicial declarada e retorna o índice do melhor indivíduo que tem maiores chances de ser selecionado.

$$F_T = \sum_{i=1}^{Nind} f_N(x_{i,j}). \quad (3.14)$$

onde:

F_T é a soma da função aptidão dos indivíduos,

$f_N(x)$ função aptidão os indivíduos da população avaliados,

$Nind$ número de indivíduos.

A partir disto, p_i é a probabilidade de seleção de cada indivíduo ser selecionado pela jogada da roleta.

$$p_i(x) = \frac{f_i(x_{i,j})}{F_T} \quad (3.15)$$

Com isto, a probabilidade de acumulação de cada indivíduo com classificação de pior aptidão em relação ao indivíduo de melhor aptidão é descrita da seguinte maneira:

$$c_i = \sum_{k=1}^i p_k, \quad i = 1, \dots, Nind \quad (3.16)$$

onde:

$\sum_{k=1}^i p_k$ é a somatória da aptidão dos indivíduos acumulada.

3.2.4 Cruzamento

É selecionado $Leoa_{i,j}$ e $Leao_{i,j}$ de melhor aptidão para gerar os filhotes, esta etapa é similar à etapa dos algoritmos genéticos referenciada cruzamento (RAJAKUMAR, 2012).

Para uma taxa de cruzamento alta, ocorre convergência prematura da população, porém quando a taxa é considerada baixa resulta no pouco aproveitamento da população. O processo de cruzamento gera novos descendentes, pois ocorre troca de informações entre os indivíduos, os pais, que são escolhidos aleatoriamente dentro da população. Para que ocorra a troca de informações dos cromossomos dos pais, é realizado um corte nesses cromossomos. Este corte pode ser representado da seguinte maneira:

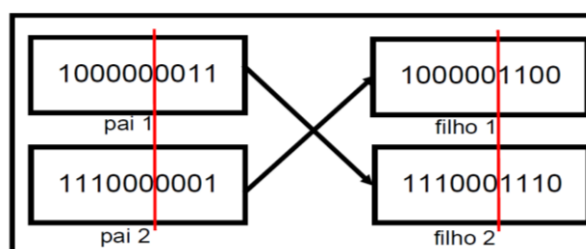


Figura 8 – Corte do cromossomo.
Fonte: Adaptada de Coelho (2003).

3.2.5 Mutação

Após realizar o corte dos cromossomos e a geração de novos descendentes a população passa pelo processo de mutação que também é similar aos AGs. O processo de mutação é adotado para a mesma formulação que os AGs utilizam para representação binária (RAJAKUMAR, 2012). A mutação altera um ou mais genes do filho escolhido a partir da taxa de mutação que é equivalente ao intervalo $[0,1]$. Mas para ocorrer esta alteração nos genes, o valor dos filhos é representado por cadeias de bits, eles podem ser descritos da seguinte maneira:

$$Filho_1 = 1000001100 = 0,5125;$$

$$Filho_2 = 1110001110 = 1,8891;$$

no momento que sofrem a mutação o gene escolhido é alterado, o que for zero passar a ser um e vice-versa. Como exemplo, se o 4º gene do $Filho_1$ e $Filho_2$ resulta em:

$$Filho_1 = 1001001100 = 0,5742;$$

$$Filho_2 = 1111001110 = 0,9473.$$

3.2.6 População nômade

A população nômade deve ser gerada através da seguinte porcentagem

$$LeaoNom_{i,j} = xAtual_{i,j} < r = 10\% ; \quad (3.16)$$

$$LeaoAtual_{i,j} = xAtual_{i,j} \geq r = 10\% . \quad (3.17)$$

A população $xAtual_{i,j}$ representa a população atualizada com os novos valores da população mutante. Assim a população nômade ($LeaoNom_{i,j}$) é gerada a partir da formulação (3.16).

3.2.7 Disputa territorial

A disputa territorial é onde serão selecionados os melhores filhotes, ou seja, o filhote de melhor aptidão e os que não representam uma boa aptidão é substituído.

A população nômade é composta pelos indivíduos masculinos que foram expulsos do grupo e voltam depois de um tempo para disputar o antigo lar. O pseudocódigo ilustra a rotina da defesa territorial que é a exclusão do $LeaoNom_{i,j}$.

Pseudocódigo 3: Disputa Territorial

Geração NovaPop = $\{\overline{x_1}, \overline{x_2}, \dots, \overline{x_{N_p}}\}$

Idade ($Filhos_n$) $\leftarrow 0$;

para i = idade ($Filhos_n$) faça até Id = 4

se $f(LeaoNom_{i,j}) < f(Leao_{i,j})$

se $f(LeaoNom_{i,j}) < (LeaoOrg_{i,j})$

Excluir: $f(xAtual_{i,j})$ e $f(x_{i,j})$;

$Leao_{i,j} \leftarrow LeaoNom_{i,j}$

se não

```

Idade (Filhosn)  $\leftarrow$  1;
fim se/senão;
fim para;

```

Após a disputa territorial, que ocorre a expulsão ou morte dos filhotes, quando o filhote é expulso do bando ele pode voltar depois de um tempo para tentar tirar o território que esta sob os cuidados $LeaoAtual_{i,j}$.

3.2.8 Aquisição territorial

A aquisição territorial se inicializa uma vez que os filhotes atingem o nível de maturidade, quando a idade dos leões é maior ou igual à idade da maturidade (id). Uma vez que os filhotes demonstram seu nível, eles podem ser considerados como os leões da população e eles começam a lutar com o $LeaoAtual_{i,j}$ para provar sua força. Caso o $LeaoAtual_{i,j}$ perca a disputa, o leão ganhador governará o bando.

A avaliação do leão e leoa determina se eles são bons o suficiente para mantê-los na população atual ou se é necessário substituí-los.

Pseudocódigo 4: Aquisição Territorial

```

Gerar:  $best_{solução} = \{Leao_{org}\}$ 
 $f(Leao_i)$  e  $f(Leoa_j) \rightarrow f(Leao_i) \neq f(Leoa_j)$ 
se  $f(Leoa_j) = Leoa_{i,j}$ 
 $\beta_i = 1$  ;
caso contrário  $\beta_i = 0$  ;
fim se;
 $Leao_{i,j} \leftarrow f(Leao_j)$ 
 $Leoa_{i,j} \leftarrow f(Leoa_j)$ 
se  $\beta_i > \beta_{força}$  faça
 $Leoa_{i,j}$  é atualizada quando for mais forte que  $Leao_{i,j}$ 
para  $f(Leoa\_Atual_{i,j}) < f(Leoa_j)$ 
 $Leoa_j \leftarrow Leoa\_Atual_{i,j}$ 
fim se
vá para o processo: cruzamento e mutação

```

faça até $id > dimensão$

A aquisição territorial é composta pelo número de reprodução das leas (β_i), ou seja, quantos filhotes foram gerados, sendo β inicializado ao gerar a população inicial do orgulho ($LeaoAtual_{i,j}$). Mas, se $Leao_{i,j}$ tiver aptidão maior que bestEMQ, ele é substituído por melhores indivíduos dos novos filhotes e a $Leao_Atual_{i,j}$, então $\beta_{força}$ se atualiza e a leoa que acaba de passar pelo processo de cruzamento e mutação é alocada na posição de $\beta_{força}$. Isto ocorrerá até $\beta_{força}$ atingir seu limite, que será do tamanho da dimensão do problema. Este processo é repetido id até atingir o número máximo dimensão. Uma vez que se atinge o limite, o melhor leão é selecionado como a melhor.

3.3 BUSCA LOCAL ITERATIVA

A principal ideia de utilizar esta metaheurística é que, quando se trata de soluções ótimas locais, fica restrito encontrar essas soluções quando processados por algoritmos de busca global (DIBBLEE et al., 2015). Com isso o algoritmo de busca local ajuda os algoritmos evolutivos a achar os ótimos locais. Assim, auxilia em obter melhorias no cromossomo da solução atual trocando no máximo $k - indivíduos$ quando ocorre a busca na vizinhança $k - permutação$ dos indivíduos (YANG et al., 2015). A busca local iterativa tem o intuito de melhorar as soluções factíveis dentro do espaço de busca determinado, realizando uma varredura ao redor das melhores soluções encontrada e grava o trajeto feito para não passar pelo mesmo mais de uma vez.

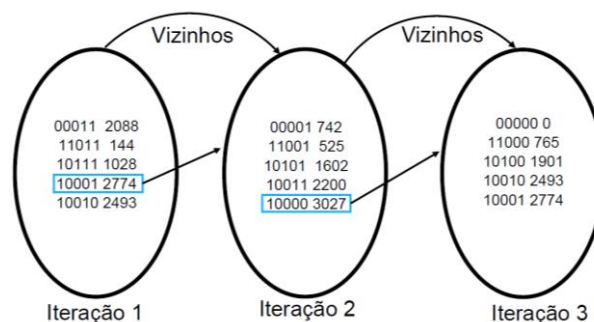


Figura 9 – Busca local por cadeia de binários.
Fonte: Adaptada de Talbi (2009).

A Figura 9 ilustra a procura de melhores soluções de ótimo locais dentro do mesmo espaço de busca entre a vizinhança. A iteração 1 obtém sua melhor solução circulada de azul, mas ao comparar com a iteração 2 acaba sendo reservada pois na iteração 2 existe um ótimo local melhor ainda. Mas ao passar pela iteração 3 não encontra um ótimo local melhor que os anteriores, por meio da estratégia de melhor vizinho é selecionado a melhor solução (TALBI, 2009).

Cada vez mais as heurísticas e metaheurísticas se tornam mais eficientes e sofisticadas, mas muitas vezes deixam de lado buscar um desempenho melhor ainda. Com isto, acabam correndo o risco de perder suas generalidades e passam a ser consideradas complexas para seu desempenho. A Busca Local Iterativa fornece uma maneira simples de satisfazer os requisitos que muitas heurísticas e metaheurísticas não foram capazes de alcançar juntas (LOURENÇO *et al.*, 2003).

3.3.1 População inicial

A população inicial é representada da maneira que a metaheurística ED realiza. Através da equação (3.1), utilizando os limites inferior e superior para gerar a população inicial.

3.3.2 Busca local

A BLI é considerada um algoritmo metaheurístico simples, porém muito robusta e poderosa. Ela aplica sua busca local na solução inicial para conseguir encontrar uma solução de ótimo local. Para a busca seguir, BLI perturba esta solução e reinicia sua busca local.

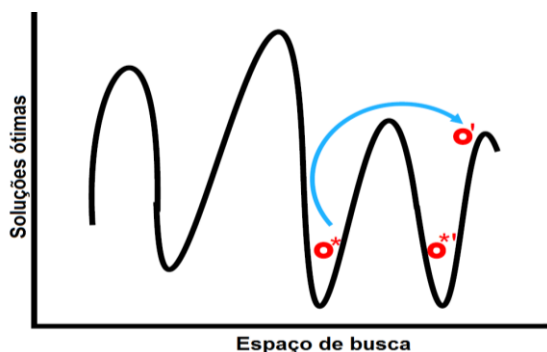


Figura 10 – Perturbação dos ótimos globais.
Fonte: Adaptada de Lourenço *et. al* (2003).

A Figura 10 mostra o mínimo local (representado de vermelho) encontrada pela função objetivo, durante a busca local a BLI tenta encontrar um mínimo local melhor do que o anterior. Neste caso, o mínimo local sofre perturbação e quando ocorre a atualização um novo mínimo é encontrado através da busca local pela vizinhança do antigo mínimo local \mathbf{o}' . A perturbação é representada pela flecha azul na Figura 10 e o novo mínimo local é \mathbf{o}^* .

A perturbação tem grande importância neste algoritmo, pois quando uma perturbação é muito pequena, ela não permite que o sistema escape da vizinhança do espaço de busca do ótimo local encontrado. Por outro lado, uma perturbação muito grande torna o algoritmo similar a uma pesquisa local de início aleatório. Mas a busca local só é válida se for capaz de obter uma boa solução ótima local, isto é, se pode encontrar dentro da vizinhança de seu espaço de busca o próximo mínimo local. Quando o espaço de busca é amplo e/ou quando a vizinhança do ótimo local é pequena, uma busca efetiva poderia ser realizada como uma trajetória apenas no conjunto de ótimo local \mathbf{o}^* , em vez do conjunto \mathbf{o} que contém todas as soluções.

Pseudocódigo 5: Busca Local Iterativa

\mathbf{o}_0 = População inicial

\mathbf{o}^* = Busca Local (\mathbf{o}_0)

Repetir;

\mathbf{o}' = Perturbação (\mathbf{o}^* , **hist**)

$\mathbf{o}^{*'} =$ Busca Local (\mathbf{o}')

$\mathbf{o}^* = \text{StopCriterio}(\mathbf{o}^*, \mathbf{o}^{*'}, \mathbf{hist})$

O pseudocódigo 5 ilustra a arquitetura dos passos realizados durante todo o processo da BLI. A população é gerada de forma aleatória com os determinados espaços de busca e pode ser representada pela equação (3.1). Esta população passa pela primeira busca atrás de ótimos locais depois que passa pelo processo de avaliação na função objetivo. Mas, a busca depende do caminho armazenado que foi realizado para chegar até o ótimo local, então é armazenado dentro da história (*hist*) da busca local. A perturbação é quantas vezes as soluções são modificadas

ao longo da rotina, pois quando se aplica, ela estará desviando-se dos ótimos locais para poder achar soluções melhores.

3.4 METAHEURÍSTICAS COLABORATIVAS

As metaheurísticas colaborativas possuem o mesmo objetivo que as metaheurísticas híbridas, juntam duas ou mais técnicas de otimização tornando-se mais robusta e com um potencial melhor para encontrar soluções factíveis do que já demonstraram individualmente. Elas trabalham de modo cascata auxiliando uma a outra para encontrar soluções factíveis quando enviam sua melhor solução para a outra/outras técnicas reavaliar em como uma tentativa de melhorar esta solução. Para tratar de problemas *Np – difíceis* é necessário o uso de técnicas de otimização flexíveis e capazes de melhorar a solução factível sem perder sua generalidade. A Figura 11 mostra a rotina de dois algoritmos trabalhando de forma colaborativa. O algoritmo da esquerda envia os melhores parâmetros estimados (*bestPar*) para o algoritmo da direita e vice-versa, desta forma cada algoritmo tentará melhorar a solução (*bestSol*) do outro algoritmo *bestPar*. A Figura 11 ilustra através das flechas azul e verde a troca de informação.

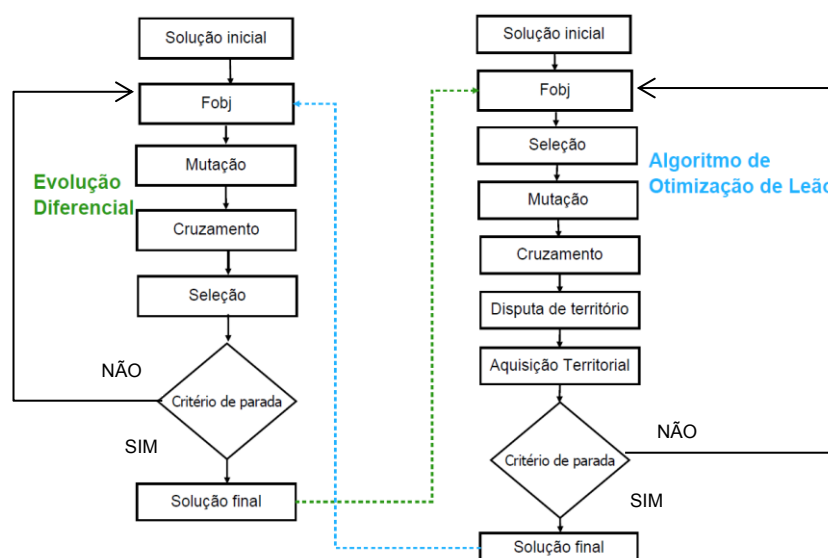


Figura 11 – Metaheurística colaborativa ED com AOL.
Fonte: Autora (2017).

O *bestPar* enviada é alocada na primeira linha da matriz que se encontra a população inicial ($x_{i,j}$) do algoritmo AOL, e consecutivamente o algoritmo ED recebe *bestPar* que também será alocada na primeira linha da matriz de $x_{i,j}$.

A Figura 12 mostra o processo colaborativo das metaheurísticas ED e BLI. Ao trabalhar de forma colaborativa, cada técnica de otimização encontra uma solução factível a partir de suas características. No caso da ED o fator de mutação (F) é a chave principal desta metaheurística, pois resulta em uma convergência robusta e rápida. Mas o algoritmo BLI, sua principal característica é a busca local ao redor de vizinhança, onde procura pela solução factível melhor.

A seta verde indica que ED envia *bestSol*, onde é substituída na primeira linha da população da BLI e, então todo o seu processo é inicializado na tentativa de melhorar *bestSol* e retornar uma solução melhor ainda para ED.

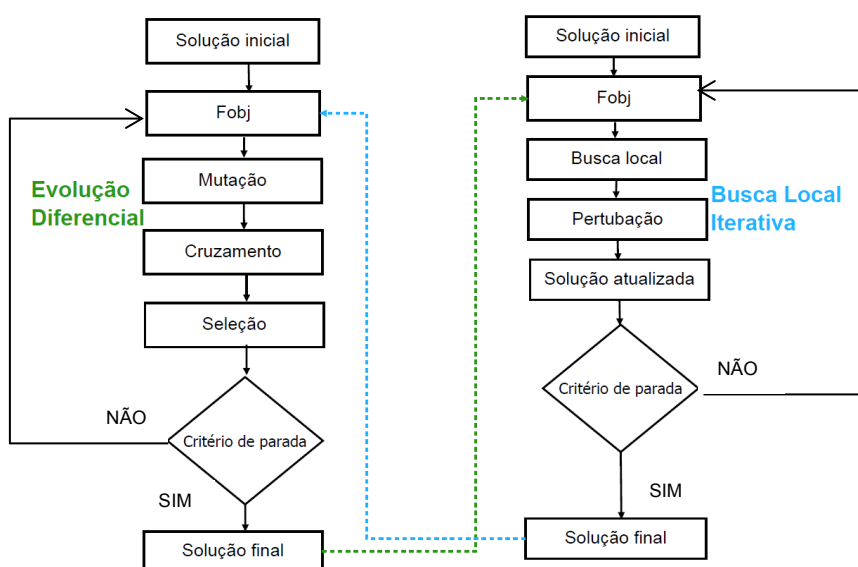


Figura 12 – Metaheurística colaborativa ED com BLI.
Fonte: Autora (2017).

A Figura 13 mostra o processo colaborativo das metaheurísticas AOL e BLI. De forma semelhante, elas trocam informações enviando *bestSol* para melhorar a solução e deixando o algoritmo mais robusto por conseguir melhorar as soluções factíveis umas das outras. Com isso o resultado dessas metaheurísticas colaborativas apresentam melhoras quando comparadas com o desempenho individual de cada metaheurística clássica.

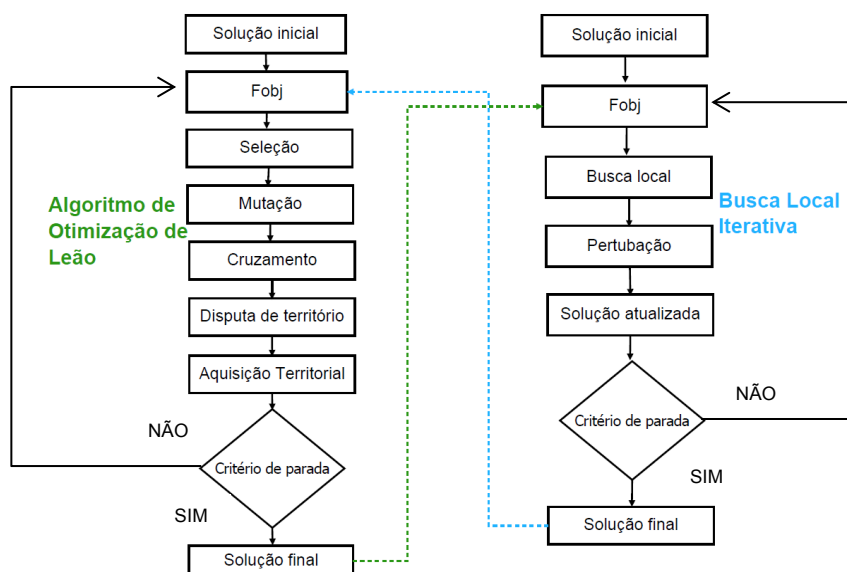


Figura 13 – Metaheur stica colaborativa AOL com BLI.
Fonte: Autora (2017).

A troca de informa  o, o envio da melhor solu  o de cada algoritmo em cada itera  o ao outro algoritmo, acelera a converg ncia das novas solu  es que s o encontradas pelo algoritmo. Desta maneira, ocorre a melhora das solu  es deixando os algoritmos eficientes e robustos em rela  o quando os algoritmos trabalham de modo individual.

4 COMPUTAÇÃO PARALELA APLICADA A METAHEURÍSTICAS

Atualmente, a computação vem tendo grandes avanços e ganhando mais ênfase no mundo das metaheurísticas. Muitas metaheurísticas levam um bom tempo computacional para encontrar soluções factíveis. Dessa maneira, vários algoritmos estão sendo desenvolvidos de forma paralela (na maioria das vezes em versões síncronas) para melhorar o desempenho em questão do tempo computacional. Esta aplicação não melhora o desempenho das metaheurísticas de modo a deixá-las mais robustas e sim, apenas aceleram seu desempenho sem perder a sua generalidade. O foco principal da paralelização das metaheurísticas se encontra em resolver instâncias de problemas complexos e grandes em um espaço de tempo que seja considerado razoável (SAIT *et al.*, 2007; ESPOSITO; GOMES, 2012).

Do ponto de vista dos algoritmos, a computação paralela explora a ordem parcial dos algoritmos de forma simultânea sem que haja modificações nos processos realizados para encontrar a melhor solução e alcançar espaços de busca maiores em um razoável tempo. No entanto, ao paralelizar as metaheurísticas, não só as acelerações têm grande importância, mas também a qualidade máxima que consegue adquirir. Portanto, para obter qualquer benefício da paralelização, é necessário não apenas uma boa partição do problema para uma distribuição uniforme de tarefas aceitável na computação. Embora o mais importante seja passar por um espaço de busca complexo considerado eficiente para poder alcançar as soluções de boa qualidade. Mas grande parte da paralelização tanto da computação de custos quanto das funções, leva em conta a interação da estratégia de paralelização com a inteligência da heurística, pois afeta diretamente a qualidade final da solução ótima, e diretamente no tempo de execução devido ao seu efeito na convergência do algoritmo. A paralelização das metaheurísticas vem sendo pesquisadas com frequência (SAIT *et al.*, 2007).

Neste capítulo é apresentado de forma sucinta às formas de paralelismo que se podem colocar as metaheurísticas e os tipos que elas podem ser enquadradas.

4.1 PARALELISMO

Geralmente, é ineficiente explorar o paralelismo de nível fácil de uma única instrução. Em vez disso, a execução simultânea de diferentes operações ocorre no nível elevado dos procedimentos ou funções. Este paralelismo é referenciado como paralelismo funcional. Quando um processo calcula o primeiro vetor derivado de uma função enquanto outro calcula a matriz de segunda derivada. Os dois processos podem ser sobrepostos parcialmente ao tempo. Porém a vantagem se encontra ao realizar cálculos complexos e de grandes dimensões. Essa sobreposição parcial pode gerar certas acelerações significativas. Os computadores paralelos hoje em dia, estão bem adaptados para executar o paralelismo funcional, que geralmente acompanham uma arquitetura MIMD onde ambos, os dados e instruções fluem simultaneamente no sistema. Mas muitas vezes os computadores MIMD são constituídos por processadores rapidamente conectados, cada um contendo um ALU e um módulo de memória. Dessa maneira, a computação paralela baseada nos dados ou paralelismo funcional é particularmente mais eficiente do que os algoritmos que controlam as estruturas de dados que são fortemente regulares, como multiplicações de matrizes (ALBA, 2005; COTTA *et al.*, 2005; SAIT *et al.*, 2007; ESPOSITO; GOMES, 2012).

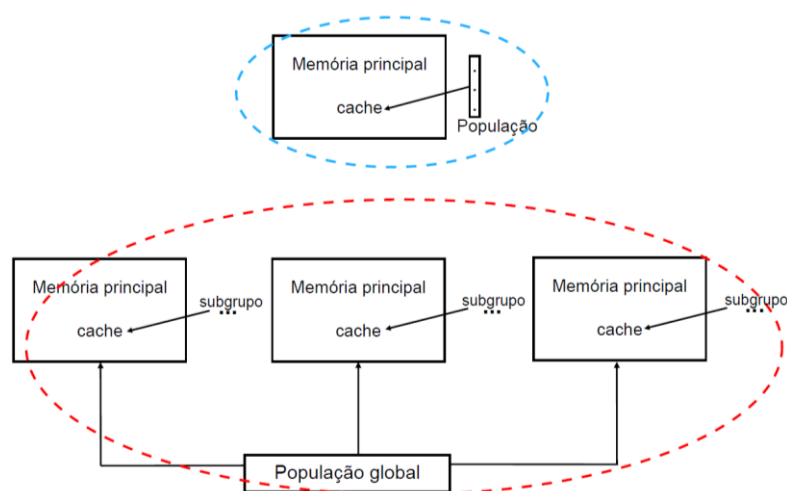


Figura 14 – Metaheurística sequencial e paralela.
Fonte: Adaptada, Alba (2005).

Quando executado em paralelo é possível perceber que a população se fragmenta em pedaços para obter a aceleração do processo, a Figura 14 ilustra este

tipo de fragmentação em relação à população. O círculo azul é uma operação sem paralelismo e quando se aplica o paralelismo, o algoritmo é dividido em fragmentos onde cada escravo assume um pedaço da população e no final do ciclo eles enviam todas as informações para forma uma única resolução.

Uma metaheurística se inicializa a partir de diferentes soluções iniciais e explora diversas regiões do espaço de busca, resultando em diferentes soluções. Estas regiões do espaço de busca podem se tornar uma fonte de paralelismo para as metaheurísticas. No entanto, a análise de desenvolvimento paralela para métodos metaheurísticos torna-se complexa pois muitas vezes o desenvolvimento paralelo não consegue retornar a mesma solução do que a do desenvolvimento sequencial. Mas, alguns critérios de avaliação baseados na qualidade da solução (ou seja, o melhor método para obter a solução ótima) são utilizados para caracterizar as medidas de aceleração. Portanto, algumas estratégias de paralelização foram desenvolvidas para serem aplicadas nas metaheurísticas de acordo com o tipo de paralelismo a ser utilizado (CRAINIC; TOULOUSE, 2003).

4.2 TIPOS DE PARALELISMO

O paralelismo do tipo 1 pode ser obtido através da execução simultânea das operações ou da avaliação simultânea de diversos movimentos que compõem uma iteração do espaço de busca. As estratégias deste tipo de paralelismo são reconhecidas por paralelismo funcional ou paralelismo de nível fácil. Elas visam reduzir diretamente o tempo de execução de uma determinada solução ótima. Quando o mesmo número de iterações é permitido para versões sequenciais e paralelas e as mesmas operações são realizadas em cada iteração (o mesmo conjunto de movimentos candidatos é avaliado e utiliza o mesmo método de seleção, como exemplo), o desenvolvimento paralelo segue o mesmo caminho de exploração do desenvolvimento sequencial resultando a mesma solução.

O paralelismo do tipo 2, vem da decomposição das variáveis de decisão de subconjuntos diferentes. Mas a metaheurística é aplicada a cada subconjunto. Assim, este tipo de paralelismo utiliza o mestre-escravo onde todo processo é dividido pelo mestre e envia os pedaços para escravo. Conforme a execução da metaheurística, os escravos exploram de modo simultâneo e independente de qual

parte do processo estiver com o escravo, ele tem acesso a uma parte do bairro (do espaço de busca) ou então a todo bairro. Assim o mestre executa uma operação mais complexa contendo soluções parciais obtidas em cada subconjunto para formar uma solução completa e eficiente para tal problema.

O paralelismo do tipo 3 é conhecido por seus diversos caminhos durante as buscas simultâneas dentro do espaço de busca das soluções onde cada linha ou encadeamento de execução (*thread*) pode ou não simular a mesma metaheurística. Com isto, conseguem inicializar diversas partes ou soluções diferentes, além de se comunicarem durante todo o processo de busca ou somente no final da execução para identificar a melhor solução entre eles. Mas para cada tipo de metaheurística é desenvolvido este tipo de paralelização, como exemplo, nos algoritmos genéticos o tamanho da população (N_p) sequencial é reduzida para cada linha de $x_{i,j}$. Mas já nos algoritmos de busca local, cada *thread* executa as iterações (T_p) onde T é o número de iterações do processo sequencial. Embora suas estratégias sejam utilizadas para fazer análises mais completas e com o uso de *multithreads* conseguem obter soluções melhores do que as metaheurísticas sequenciais, mesmo que o tempo de cada segmento é menor do que a computação sequencial.

Dentre dessas três técnicas de paralelização, o presente trabalho utiliza a primeira técnica que pode ser ilustrada pelo seguinte pseudocódigo:

Pseudocódigo 6: Paralelização

$x_{i,j}$ inicializado aleatoriamente;
 Para $i \leftarrow 1$ até $n - \textit{iteração}$ faça
 $\bar{x}_j = \textit{argmin } f(x_{i,j}) = \{x_1, x_2, \dots, x_j\};$
 fim i ;

Este tipo de paralelização é aplicado na função de custo, por acelerar os cálculos da função a cada execução de $n - \textit{iteração}$ e resultando na melhora do tempo gasto a cada iteração por cada algoritmo. Uma vez que, problemas $Np - \textit{difíceis}$ resultam no retardamento dos algoritmos devido a complexidades dos sistemas não lineares.

5 RESULTADOS

Para a obtenção dos resultados foram utilizados três estudos de caso, sendo eles: função teste de Ackley e Rosenbrock; estimação de parâmetros do conjunto conduto/turbina utilizando dados de uma turbina Francis coletados em campo; e estimação de parâmetros de um sistema de controle de velocidade (malha hidráulica e conduto/turbina) de uma Usina Hidrelétrica, com turbina Kaplan, utilizando dados coletados em campo. Os dados de campo foram obtidos de usinas hidroelétricas pertencentes ao Sistema Interligado Nacional (SIN), localizadas na região sudeste do Brasil.

5.1 PROBLEMAS TESTES DE OTIMIZAÇÃO

O primeiro estudo de caso foi utilizado para uma avaliação das metaheurísticas através de funções testes de otimização, e para a validação do desempenho de cada algoritmo.

Price *et al.* (1997) citam algumas destas funções testes, eles afirmam que elas podem ser divididas em três categorias: unimodal sem restrições; multimodal sem restrições; e multimodal com restrições. A maioria destas funções pode tratar de problemas *Np – difíceis* ou de dimensões pequenas. Quanto maior for a dimensão mais irá se exigir do algoritmo em uso, pois aumenta o esforço computacional retardando a sua convergência. Cada função depende do conjunto de limites de parâmetros iniciais superiores (B_u) e inferiores (B_l). Na maioria dos casos, o mínimo é independente da dimensão da função, mas se o mínimo da função objetivo depender da dimensão, então os valores da função ideal para as dimensões selecionadas são fornecidos para tais problemas. Deste modo, foram escolhidas duas funções de teste, Ackley e Rosenbrock, ambas tratam de problemas complexos.

Tabela 1 – Configuração das funções de teste.

Função de Teste	dim	B_u	B_l	Mín. global
Ackley	100	32	-32	$f(x^*) = 0$
Rosenbrock	100	10	-5	$f(x^*) = 0$ ou $\neq 0$

Fonte: Autora (2018).

A função Ackley utiliza gráfico bidimensional para mostrar a localização das soluções ótimas locais, pois é composta de uma região plana e canal no centro. Sua formulação pode ser representada por:

$$f(x) = -a \exp\left(-b \sqrt{\frac{1}{d} \sum_{i=1}^{dim} x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^{dim} \cos(cx_i)\right) + a + \exp(1) \quad (5.1)$$

onde:

$$a = 20,$$

$$b = 0,2,$$

$$c = 2\pi,$$

dim é a dimensão do problema,

x_i é o indivíduo selecionado da população inicial ($x_{i,j}$).

Para a função Rosenbrock o espaço de busca é dentro do intervalo $[-32,32]$ e esta função trata com facilidade de problemas *Np – difíceis* resultando em boas soluções. O mínimo global está localizado no vale plano, que é estreito e parabólico. Sua representação matemática é descrita pela equação (5.2) a seguir.

$$f(x) = 10dim + \sum_{i=1}^{dim} [x_i^2 - 10\cos(2\pi x_i)] \quad (5.2)$$

onde:

dim é a dimensão;

x_i são os indivíduos da população inicial selecionados aleatoriamente.

As equações (5.1) e (5.2) resultam em soluções ótimas, com todos os algoritmos possuindo as seguintes características: o tamanho da população igual a 100 ($N = 100$); o tamanho da dimensão é igual a cem ($dim = 100$); máximo de dez mil iterações ($iter = 10000$); com critério de parada 10^{-3} ($stopCritério = 0,001$); e duzentas e cinquenta realizações ($count = 500$) ao total.

As configurações dos algoritmos são representadas pela Tabela 2, onde $estr_F$ é a estratégia de mutação, F é o valor da taxa de mutação, Cr é o valor da taxa de cruzamento, Id é a idade adulta dos leões, $tamBusca$ tamanho da busca para encontrar as melhores soluções, P é a perturbação que a melhor solução irá sofrer durante a $tamBusca$. Os valores de F e Cr foram escolhidos a partir dos testes

realizados e que demonstraram boa convergência com estes valores, resultando em ótimas soluções.

Tabela 2 – Configuração das metaheurísticas

Algoritmo	<i>estr_F</i>	<i>F</i>	<i>Cr</i>	<i>Id</i>	<i>tamBusca</i>	<i>P</i>
ED	DE/best/1	0,2 - 0,8	0,7	-	-	-
AOL	Binário	0,3	0,6	4	-	-
BLI	-	-	-	-	45	$N/2$

Fonte: Autora (2018).

Com os dados da Tabela 2, obteve-se os resultados valores mínimo, máximo, médio e desvio padrão (Σ) encontradas pelas metaheurísticas de forma sequencial. Por fim, para uma análise mais justa das soluções, a população inicial é igual para todos os algoritmos.

Tabela 3 – Variação das soluções de Ackley.

Metaheurísticas	Mín.	Max	Méd.	Σ
Evolução Diferencial (ED)	2,087	5,056	3,507	0,902
Algoritmo de Otimização de Leão (AOL)	0,0045	10,741	4,217	3,180
Busca Local Iterativa (BLI)	0	2,865	1,85	0,219
ED com AOL	0,0058	19,719	10,501	6,6123
ED com BLI	0	2,8927	1,7126	1,2134
AOL com BLI	0	2,8657	1,7857	1,1660

Fonte: Autora (2018).

Analisando a Tabela 3, observa-se que a metaheurística BLI é a que tem a menor variação, o que significa que os resultados encontrados por ela estão sempre próximos uns dos outros (repetibilidade). Já a metaheurística AOL embora tenha uma maior variação do que a BLI, é a segunda melhor, pois fica bem próxima do mínimo global da função teste de otimização Ackley. A BLI e AOL encontram facilmente a solução ótima, o que mostra a sua agilidade em lidar com problemas *Np – difíceis*. Quanto a ED para este problema foi a que mostrou a menor variação, o que significa que as soluções encontradas são boas também.

A Figura 15 ilustra a variação obtida pelas soluções durante as 250 realizações.

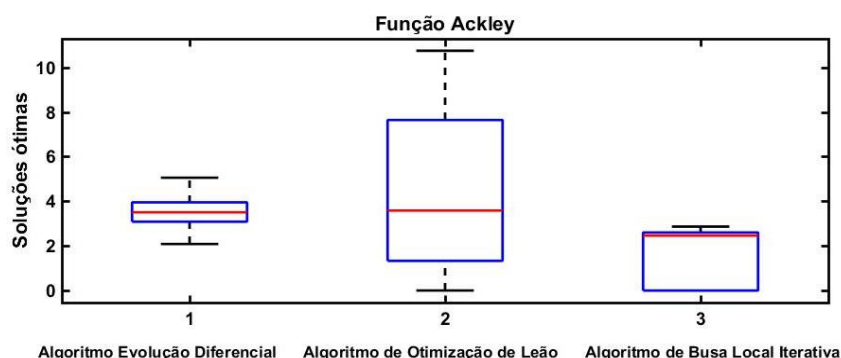


Figura 15 – Ackley com metaheurísticas individuais.
Fonte: Autora (2018).

A linha vermelha de caixa apresentada na Figura 15 significa o valor médio das soluções factíveis, quanto maior é a caixa maior, ou seja, maior é o desvio padrão (Σ) das soluções encontradas por cada algoritmo.

Para as metaheurísticas colaborativas a Figura 16 ilustra a variação obtida pelas metaheurísticas.

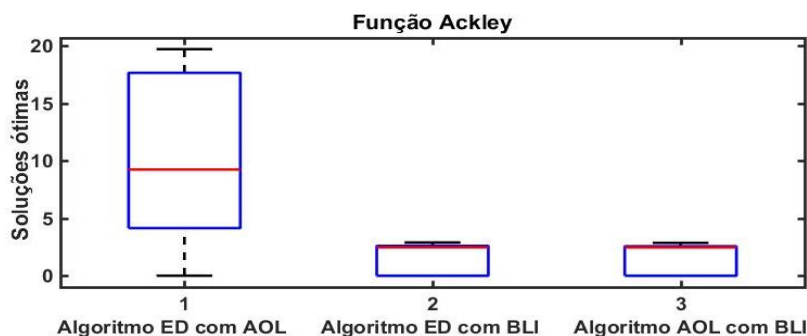


Figura 16 – Ackley com metaheurísticas colaborativas.
Fonte: Autora (2018).

Quando as metaheurísticas trocam informações, ocorre a diminuição do Σ . A diminuição ocorreu nas colaborativas: ED com BLI; e AOL com BLI. Embora as soluções factíveis para os algoritmos colaborativos, ED com BLI e AOL com BLI, apresentam o valor médio e a variância desses algoritmos colaborativos é menor em relação aos algoritmos individuais. Mas, tanto os algoritmos individuais e colaborativos, eles resultam no mínimo global da função ou na solução factível bem próximo.

Para a segunda função teste de otimização, Rosenbrock, os resultados obtidos começam com valores altíssimos (10^4) e aproximam-se do mínimo global da função. A Tabela 4 ilustra os respectivos valores médios, mínimos, máximos e desvio padrão (Σ) durante as 500 realizações.

Tabela 4 – Variação das soluções de Rosenbrock.

Metaheurísticas	Mín.	Máx.	Méd.	Σ
Evolução Diferencial (ED)	9,027	730,50	157,542	118,17
Algoritmo de Otimização de Leão (AOL)	0,0059	$3,623e^4$	$3,83e^3$	$6,517e^3$
Busca Local Iterativa (BLI)	0	17,174	4,787	4,285
ED com AOL	0,0051	$2,397e^4$	$268,5e^3$	$1,64e^3$
ED com BLI	0	17,0430	4,6104	4,2808
AOL com BLI	0	19,078	4,5227	4,3195

Fonte: Autora (2018).

Analisando a Tabela 4, observa-se que a técnica colaborativa consegue melhorar as soluções encontradas pelas metaheurísticas individualmente, resultando em aproximações ou valores exatos do mínimo global da função. Para uma análise visual, a Figura 17 ilustra as metaheurísticas individuais e a Figura 19 as metaheurísticas colaborativas.

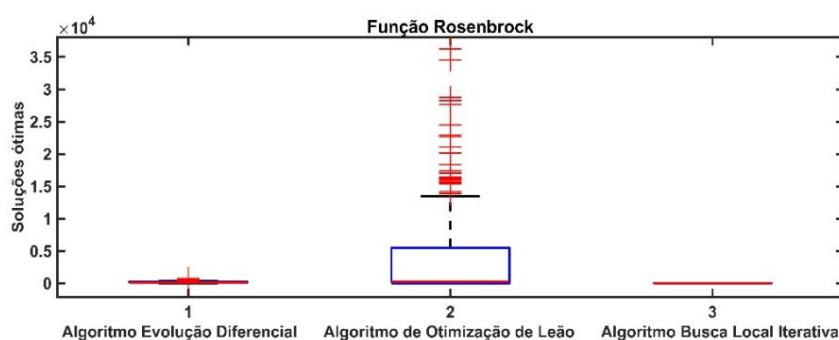


Figura 17– Rosenbrock com metaheurísticas individuais.

Fonte: Autora (2018).

Fica visível na Figura 18, que o algoritmo BLI obteve as melhores soluções e também a menor variação (repetibilidade), alcançando o mínimo global da função muitas vezes. Já no caso dos algoritmos colaborativos, ocorrem as melhores

soluções quando o algoritmo BLI está presente evidenciando sua participação na obtenção da solução ótima.

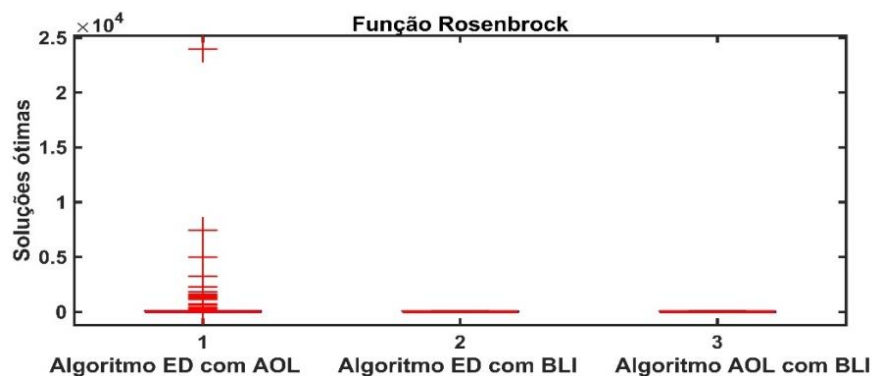


Figura 18 – Rosenbrock com metaheurísticas colaborativas
Fonte: Autora (2018)

Estas duas funções testes de otimização validam os algoritmos, pois eles conseguem obter o mínimo global das duas funções. Por fim, uma análise do desempenho do tempo computacional dessas metaheurísticas demonstrou rapidez ao lidar com dimensão grande ($dim = 100$) quando estas foram paralelizadas.

Tabela 5 – Tempo de processamento

Metaheurísticas	Sequencial	Paralelo
Evolução Diferencial (ED)	10 min e 51 s	50,1 s
Algoritmo de Otimização de Leão (AOL)	6 min e 28 s	41,5 s
Busca Local Iterativa (BLI)	10 min e 04 s	49 s
ED com AOL	9 min e 46 s	1 min e 48 s
ED com BLI	10 min e 50 s	1 min e 43 s
AOL com BLI	9 min e 38 s	1 min e 20 s

Fonte: Autora (2018).

O tempo computacional gasto por estas metaheurísticas são razoáveis quando sequenciais, mas quando são paralelizadas pode se notar uma diminuição do tempo de processamento o que as torna atrativas para aplicações em tempo real em várias áreas da ciência.

Após a obtenção dos valores das soluções, o tempo de processamento foi avaliado e não sendo muito diferente da função de teste Rosenbrock. O tempo de

processamento dessa função é mostrado na Tabela 6 onde foi coletado a partir do comando *tic-toc* da plataforma Matlab®.

Tabela 6 – Tempo de processamento

Metaheurísticas	Sequencial	Paralelo
Evolução Diferencial (ED)	3 min e 30 s	51,4 s
Algoritmo de Otimização de Leão (AOL)	15 min e 11 s	1 min e 02 s
Busca Local Iterativa (BLI)	3 min e 33 s	52 s
ED com AOL	8 min e 10 s	1 min e 07 s
ED com BLI	8 min e 49 s	1 min e 16 s
AOL com BLI	9 min e 09 s	1 min e 21 s

Fonte: Autora (2018)

O tempo de processamento destas funções de testes são significativamente rápidas, pois seus cálculos são simples. Porém para aplicações de problemas do nosso cotidiano, alguns sistemas reais que trabalham geralmente com grandes escalas acabam tendo um atraso significativo no processamento computacional devido à quantidade de seus dados que são utilizados nas metaheurísticas.

5.2 CONJUNTO CONDUTO/TURBINA HIDRÁULICA

O segundo estudo de caso, trata do problema da estimação de parâmetros do sistema conjunto conduto/turbina, e os dados utilizados na estimação é de uma turbina do tipo Francis obtidos em ensaios de campo.

As turbinas hidráulicas do tipo Francis são consideradas turbinas de reações, de pás fixas que controlam a passagem da água e opera entre quedas de 40 m até 400 m, sua curva de rendimento plana garante um bom desempenho em uma ampla faixa de operação.

O desempenho de uma turbina hidráulica depende do controlador de velocidade, onde ele é a chave principal do sistema. A partir disso, consegue-se controlar a potência e frequência do sistema gerador representado na Figura 19. O controlador de velocidade acionada os palhetas através de uma malha hidráulica, supervisionando a abertura das pás de admissão de água para a turbina hidráulica (KISHOR et al., 2007).

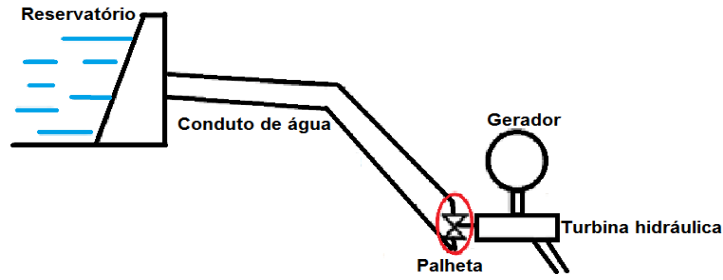


Figura 19 – Diagrama da turbina hidráulica.
Fonte: Kishor, Saini, Singh (2007).

O modelo utilizado nas simulações é uma adaptação do modelo proposto pelo IEEE (IEEE, 1992), e desenvolvido no trabalho de Osinski (OSINSKI, 2017). Este modelo é do tipo caixa cinza, pois os dados de entrada, saída e a estrutura física são conhecidos.

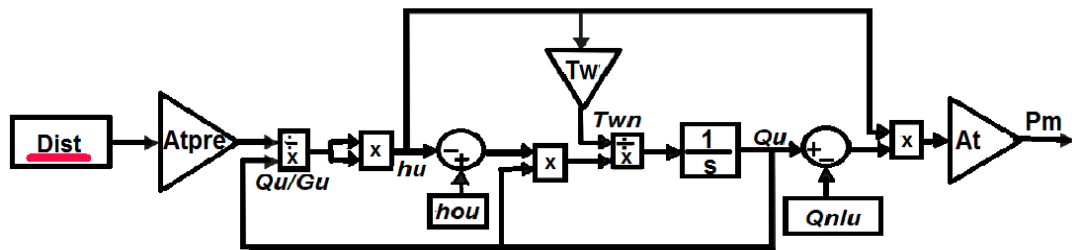


Figura 20 – Modelo conjunto conduto/turbina.
Fonte: Autora (2018).

Da Figura 20 tem-se que, o sinal de saída é a potência mecânica (P_m) da turbina Francis em pu, o sinal de entrada é a abertura do distribuidor (D_{ist}) em pu, G_u é a posição do *gate* do distribuidor em pu, Q_u é a vazão em pu, h_u é a altura em pu, hou é o valor de estado estacionário inicial de ho em pu, T_w é a constante de tempo de inércia da água em segundos, Q_{nlu} é a vazão necessária para suprir as perdas em vazio da turbina em pu, $1/s$ é o integrador; A_{tpre} , A_t são os respectivos ganhos do sistema.

Neste estudo de caso, foi definido o tamanho da população igual a trinta ($N = 30$); dimensão é a quantidade de parâmetros a ser estimados, no caso são três parâmetros ($dim = 3$); quinhentas iterações ($iter = 500$); critério de parada igual a 10^{-3} ($stopCriteria = 0,001$); e quinhentas realizações ($count = 500$). A Tabela 7 mostra as configurações de cada algoritmo.

Tabela 7 – Configuração das metaheurísticas

Algoritmo	<i>estr_F</i>	<i>F</i>	<i>Cr</i>	<i>Id</i>	<i>tamBusca</i>	<i>P</i>
ED	DE/best/1	0,2 - 0,8	0,7	-	-	-
AOL	Binário	0,3	0,6	4	-	-
BLI	-	-	-	-	5	$Np/2$

Fonte: Autora (2018).

A taxa de mutação é definida por *F*, a taxa de cruzamento por *Cr*, a idade adulta dos leões é *Id*, o tamanho da busca local é *tamBusca* e a perturbação aplicada nas soluções ótimas é *P*.

Para os parâmetros estimados, foram considerados como restrições os seguintes valores:

Tabela 8 – Configuração das metaheurísticas

Parâmetros	Restrições
A_{tpre}	0,5 a 0,9
T_w	1,7 a 2
A_t	1,3 a 1,6

Fonte: Autora (2018).

As configurações apresentadas são aplicadas nas metaheurísticas de modo sequencial e paralelo para a obtenção dos resultados.

5.2.1 Metaheurísticas rodadas de forma sequencial.

Com *count* = 500 foi possível obter os valores mínimo, máximo, média e desvio padrão de todos os parâmetros estimados pelos algoritmos individuais e colaborativos rodados de forma sequencial, os quais são mostrados na Tabela 10.

Tabela 10 – Parâmetros estimados pelas metaheurísticas sequencialmente.

Metaheurística ED	Mín.	Máx.	Méd.	Σ
A_{tpre}	0,508	0,999	0,796	0,907

T_w	1,501	1,999	1,897	0,219
A_t	1,231	1,983	1,528	0,198
Metaheurística AOL	Mín.	Máx.	Méd.	Σ
A_{tpre}	0,5092	0,919	0,774	0,091
T_w	1,507	1,99	1,827	0,156
A_t	1,258	1,778	1,5159	0,093
Metaheurística BLI	Mín.	Máx.	Méd.	Σ
A_{tpre}	0,539	0,997	0,7895	0,053
T_w	1,559	1,994	1,810	0,144
A_t	1,299	1,672	1,457	0,064
Metaheurística ED com AOL	Mín.	Máx.	Méd.	Σ
A_{tpre}	0,759	0,889	0,807	0,045
T_w	1,50	1,993	1,889	0,158
A_t	1,30	1,556	1,5021	0,092
Metaheurística ED com BLI	Mín.	Máx.	Méd.	Σ
A_{tpre}	0,5661	0,999	0,801	0,089
T_w	1,5102	1,999	1,885	0,195
A_t	1,3155	1,955	1,532	0,108
Metaheurística AOL com BLI	Mín.	Máx.	Méd.	Σ
A_{tpre}	0,550	0,893	0,787	0,010
T_w	1,502	1,978	1,899	0,110
A_t	1,2450	1,598	1,543	0,0361

Fonte: Autora (2018).

A variância de todos os parâmetros é demonstrada por Σ , para visualizar melhor, a Figura 21 representa a variação dos parâmetros estimados por cada metaheurística individualmente.

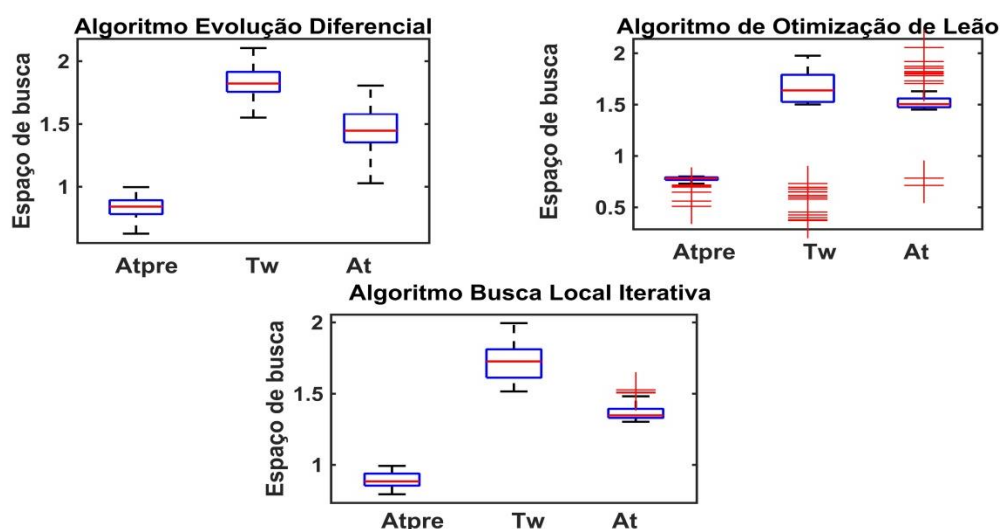


Figura 21 – Parâmetros das metaheurísticas individuais sequencialmente.
Fonte: Autora (2018).

Ao comparar as três metaheurísticas individualmente, os parâmetros estimados pelas metaheurísticas ED e BLI mostraram ter uma variação menor, ou seja, desvio padrão menor em relação à metaheurística AOL.

Mas, ao observar o valor médio dos três métodos apresentados na Tabela 10, verifica-se que os valores estão muito próximos uns dos outros.

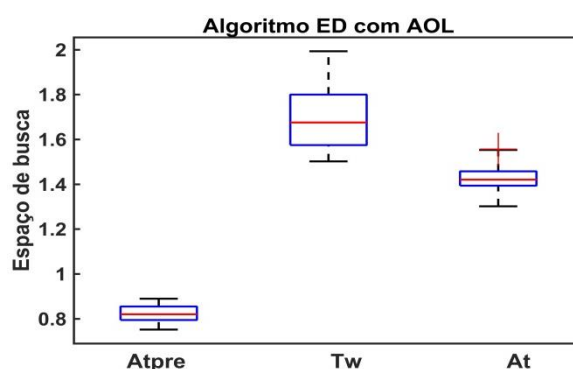


Figura 22 – Parâmetros estimados por ED com AOL sequencialmente.
Fonte: Autora (2018).

As soluções factíveis dos algoritmos colaborativos apresentam valores médios próximos uns dos outros e também com os algoritmos individuais. Pode se observar no tamanho das caixas da Figura 22, que há uma maior repetibilidade dos

valores dos parâmetros, sendo elas menores do que as caixas apresentadas pela Figura 21 que representa a metaheurística AOL e ED de forma individual.

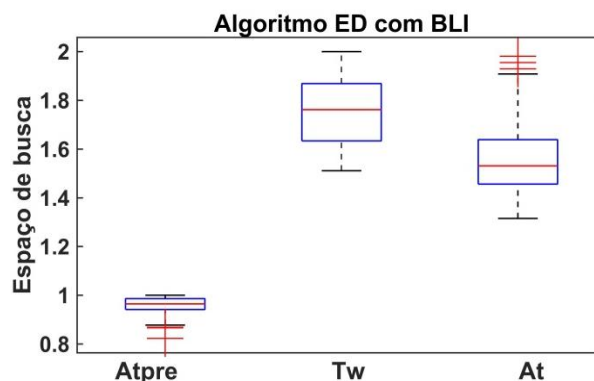


Figura 23 – Parâmetros estimados por ED com BLI sequencialmente.
Fonte: Autora (2018).

Mas quando a metaheurística ED troca informação com BLI acaba correndo variação dos parâmetros um pouco maior, do que as metaheurísticas na versão individual (clássica).

Por fim, a Figura 24, ilustra as metaheurísticas colaborativas AOL com BLI e pode-se perceber que obteve menor variação dos três parâmetros estimados.

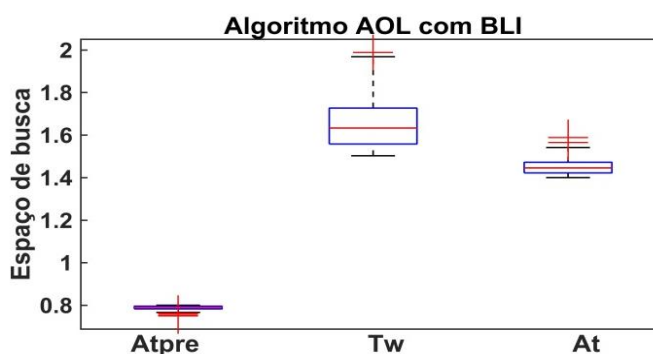


Figura 24 – Parâmetros estimados por AOL com BLI sequencialmente.
Fonte: Autora (2018).

Com as Figuras 22 a 24, a variância dos parâmetros mostra que a metaheurística colaborativa AOL com BLI é menor ao comparar com as outras metaheurísticas colaborativas.

As Figuras 25 e 26 mostram as saídas obtidas com o modelo utilizando os valores médios de cada metaheurística de forma individual e colaborativa.

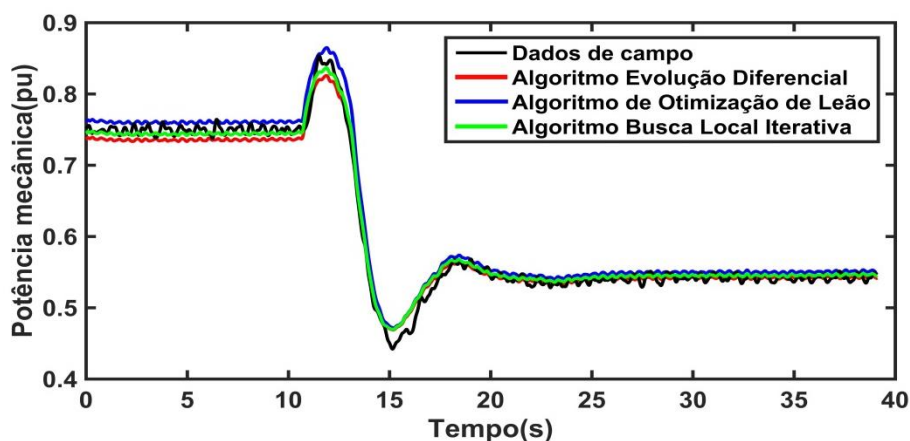


Figura 25 – Saída das metaheurísticas individuais sequencialmente.
Fonte: Autora (2018).

A Figura 25 ilustra a saída do modelo obtida dos parâmetros estimados pelos algoritmos individuais com a saída obtida por dados de medições, e pode se verificar que as saídas estão quase sobrepostas uma das outras.

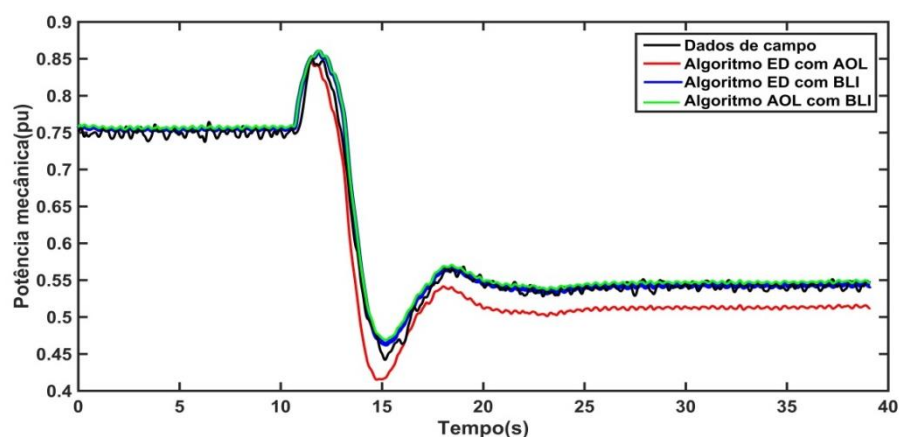


Figura 26 – Saída das metaheurísticas colaborativas sequencialmente.
Fonte: Autora (2018).

Já para a saída do modelo dos algoritmos colaborativos, a Figura 26 mostra a aproximação com a saída obtida por dados de medições. Porém, o algoritmo colaborativo ED com AOL apresenta uma curva mais afastada, isto é devido aos parâmetros estimados. Mas, quanto aos dois algoritmos colaborativos, resultam em uma saída melhor.

Para uma análise comparativa das metaheurísticas é utilizado o parâmetro estatístico erro médio quadrático (EMQ), dado por:

$$EMQ = \sum_{i=1}^{iter} (|a_i^{Ref} - a_i^{est}|)^2 \quad (5.3)$$

onde:

a_i^{Ref} é a resposta de referência,

a_i^{est} é a resposta obtida pelas metaheurísticas,

$iter$ é o número máximo de iterações.

Desta maneira, a Tabela 11 demonstra os respectivos valores obtidos através da equação (5.3).

Tabela 11 – Variância do EMQ dos algoritmos individuais sequenciais.

Comparação	ED	AOL	BLI
Mínimo	15,720	79,0267	37,615
Máximo	158,417	152,892	150,661
Média	20,797	90,716	75,7487
Desvio padrão (Σ)	9,651	12,465	15,721

Fonte: Autora (2018).

Para os algoritmos colaborativos, a Tabela 12 apresenta seus respectivos valores alcançados.

Tabela 12 – Variância do EMQ dos algoritmos colaborativos sequenciais.

Comparação	ED com AOL	ED com BLI	AOL com BLI
Mínimo	47,758	5,3285	73,668
Máximo	119,931	65,1726	155,5003
Média	75,7487	15,3013	90,7689
Desvio padrão (Σ)	14,6369	8,4728	9,7921

Fonte: Autora (2018).

Analisando a tabela 11, a metaheurística ED apresenta melhor variância em relação aos demais algoritmos individuais. Para as metaheurísticas colaborativas, a ED com BLI apresenta o melhor EMQ, pois sua variância é menor do que as outras técnicas colaborativas e individuais.

Para uma análise melhor, as Figuras 27 e 28 mostram a variância de cada uma das técnicas utilizadas.

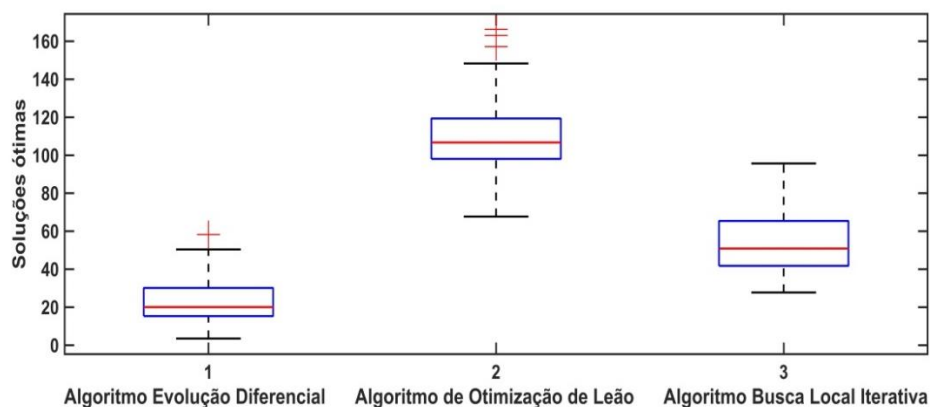


Figura 27 – EMQ das metaheurísticas individuais sequencialmente.
Fonte: Autora (2018).

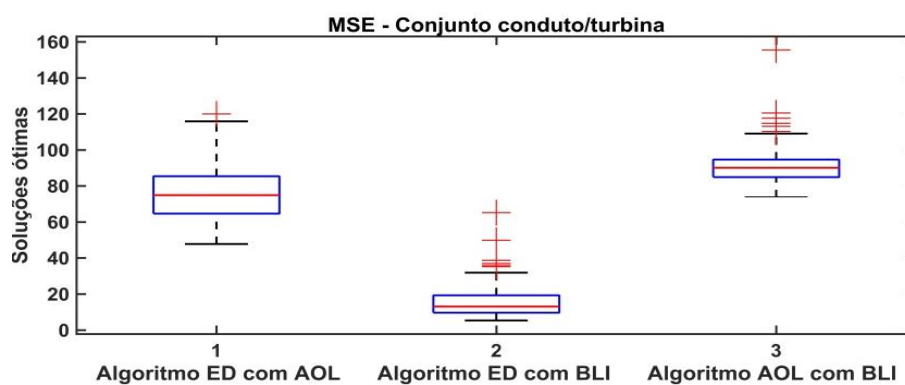
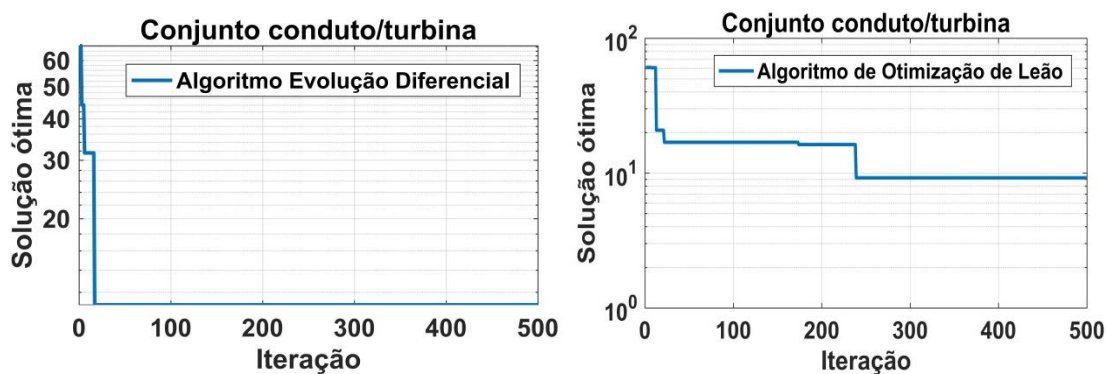


Figura 28 – EMQ das metaheurísticas colaborativas sequencialmente.
Fonte: Autora (2018).

Na Figura 29 mostra – se o valor da solução a cada iteração de cada algoritmo, demonstrando a convergência durante toda a execução.



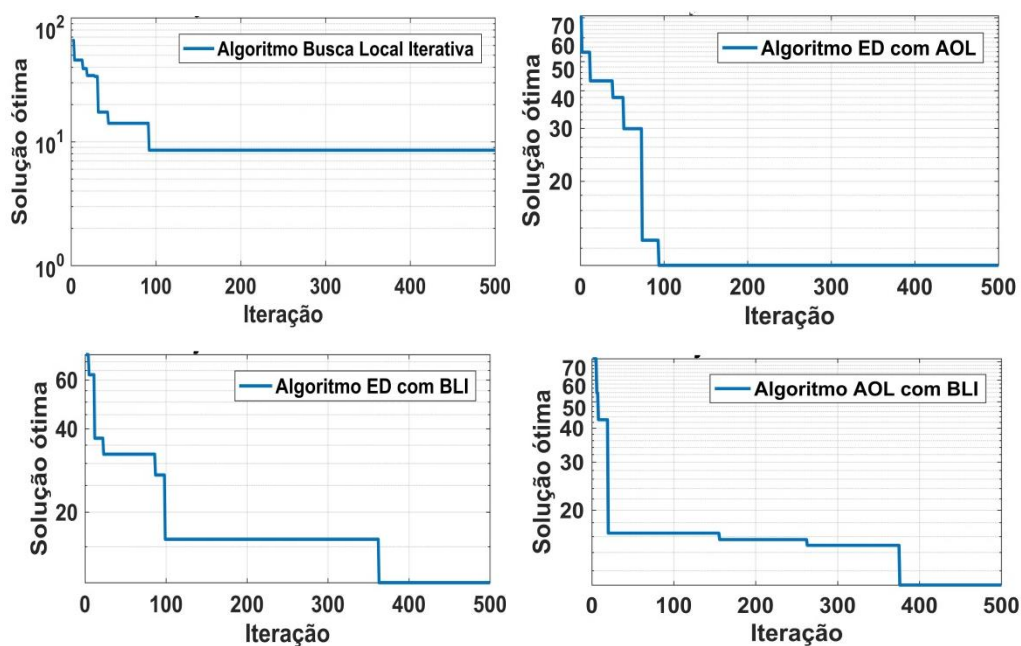


Figura 29 – Convergência das soluções.
Fonte: Autora (2018).

5.2.2 Metaheurísticas rodadas de forma paralela.

A Tabela 13 mostra os parâmetros estimados pela metaheurísticas paralelizadas, e pode-se perceber que o desvio padrão (Σ) melhora quando é aplicado o paralelismo, pois além de acelerar todo o processo apresentando uma convergência mais rápida e eficiente.

Tabela 13 – Parâmetros estimados pelas metaheurísticas paralelizadas.

Metaheurística ED	Mín.	Máx.	Méd.	Σ
A_{tpre}	0,5120	0,9374	0,7185	0,0732
T_w	1,7841	1,5092	1,8251	0,1012
A_t	1,2142	1,6544	1,4469	0,1291
Metaheurística AOL	Mín.	Máx.	Méd.	Σ
A_{tpre}	0,5087	0,9981	0,768	0,046
T_w	1, 7724	1,9998	1,86	0,4213
A_t	1,212	1,6999	1,518	0,1663
Metaheurística BLI	Mín.	Máx.	Méd.	Σ
A_{tpre}	0,5932	0,9530	0,7932	0,0510

T_w	1,7899	1,9999	1,7192	0,1163
A_t	1,3021	1,6995	1,4665	0,0558
Metaheurística ED com AOL	Mín.	Máx.	Méd.	Σ
A_{tpre}	0,5770	0,8893	0,7703	0,0359
T_w	1,7023	1,9931	1,8480	0,1324
A_t	1,2018	1,6528	1,4942	0,0503
Metaheurística ED com BLI	Mín.	Máx.	Méd.	Σ
A_{tpre}	0,5039	0,9956	0,7789	0,0297
T_w	1,7021	1,9997	1,8991	0,0193
A_t	1,2074	1,6698	1,5099	0,0574
Metaheurística AOL com BLI	Mín.	Máx.	Méd.	Σ
A_{tpre}	0,5099	0,9788	0,7712	0,1083
T_w	1,7159	1,9291	1,8009	0,0644
A_t	1,2522	1,6078	1,5195	0,2153

Fonte: Autora (2018).

O paralelismo melhora a busca de soluções factíveis e o tempo de processamento do algoritmo. A melhora pela busca de soluções está em acelerar o cálculo da função objetivo fazendo que o algoritmo encontre rapidamente a cada iteração boas soluções. Pois algumas iterações ficam procurando por uma solução factível e acaba resultando no retardamento do processo por não encontrar uma solução que satisfaça o critério de parada do algoritmo.

Tabela 14 – Variação do EMQ dos algoritmos individuais paralelos.

Comparação	ED	AOL	BLI
Mínimo	4,8019	27,7317	67,6669
Máximo	71,8800	95,6457	166,2288
Média	18,2122	53,8581	109,834
Desvio padrão (Σ)	8,8420	16,8366	18,9167

Fonte: Autora (2018).

Analizando a Tabela 14, observa-se que o algoritmo ED possui a melhor variância do que os outros algoritmos. Já para os algoritmos colaborativos, a Tabela 15 ilustra os valores alcançados.

Tabela 15 – Variação do EMQ dos algoritmos colaborativos paralelos.

Comparação	ED com AOL	ED com BLI	AOL com BLI
Mínimo	47,7590	2,2599	1,5981
Máximo	105,2745	10,1554	9,4429
Média	73,6995	3,8894	3,4362
Desvio padrão (Σ)	14,6956	2,7476	2,3557

Fonte: Autora (2018).

A metaheurística colaborativa AOL com BLI apresenta ser a melhor dentre as técnicas colaborativas, pois possui menor desvio padrão e média.

Com os resultados mostrados nas Tabelas 14 e 15, resultam nas Figuras 30 e 31, para uma análise visual da variância dos valores de todos os algoritmos.

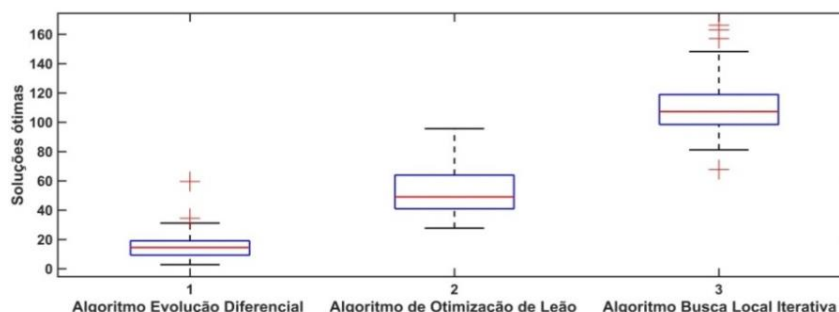


Figura 30 – EMQ das metaheurísticas individuais paralelizadas.

Fonte: Autora (2018).

Percebe-se que a solução factível do algoritmo BLI ultrapassa o valor de 160, enquanto os outros algoritmos conseguem soluções factíveis de até 100. Quando estes algoritmos trabalham de forma colaborativa (trocando informações), as chances de se encontrar soluções de menor variância são maiores.

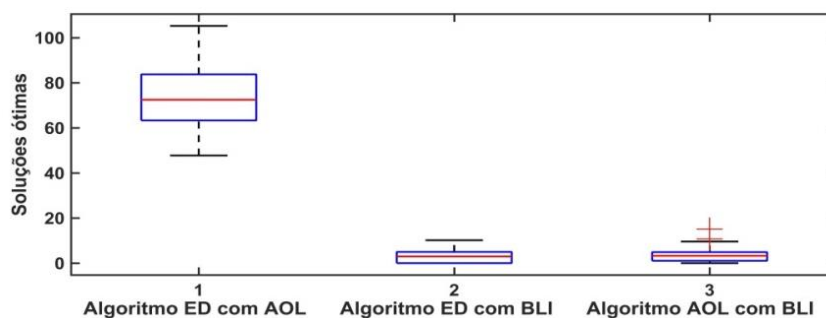


Figura 31 – EMQ das metaheurísticas colaborativas paralelizadas.
Fonte: Autora (2018).

Para a variação do EMQ das metaheurísticas paralelizadas, foram obtidos valores menores do que os algoritmos sequenciais. Pois, o paralelismo funcional consegue melhorar o cálculo da função objetivo, que consecutivamente resultam em valores melhores para as soluções.

A saída do modelo identificado com os parâmetros estimados por cada algoritmo resulta em uma aproximação ao ser comparada com a saída dos dados em campos e apresentam ser melhores do que os parâmetros estimados com os algoritmos sequenciais.

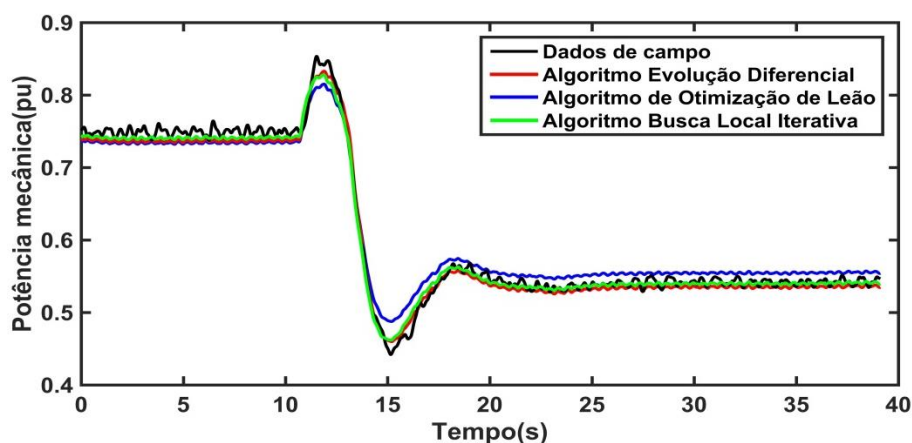


Figura 32 – Saída das metaheurísticas individuais paralelizadas.
Fonte: Autora (2018).

Percebe-se que ao comparar a Figura 32 com a Figura 26, que há uma aproximação melhor e nenhuma das saídas obtidas pelos algoritmos passam a saída de referência. Para as metaheurísticas colaborativas paralelizadas, também resultam em melhoras ao comparar as saídas das metaheurísticas colaborativas

paralelizadas com as metaheurísticas colaborativas sequenciais. A Figura 33 ilustra a saída das metaheurísticas colaborativas paralelizadas.

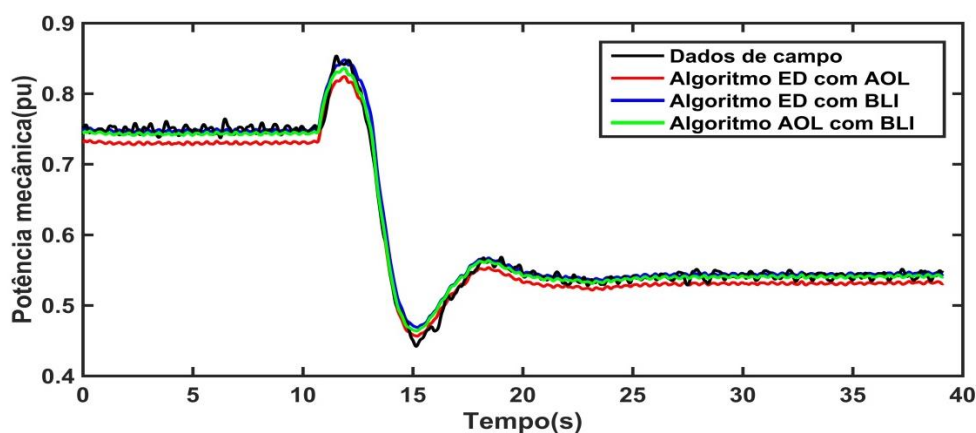


Figura 33 – Saída das metaheurísticas colaborativas paralelizadas.
Fonte: Autora (2018).

Por fim, no que diz respeito ao tempo de processamento de cada algoritmo é apresentado na Tabela 16 os seus valores. Observa-se que o tempo é obtido com cada metaheurística sendo executada de forma sequencial e paralela, e o tempo foi medido através do comando *tic-toc* da plataforma Matlab®.

Tabela 16. Tempo de processamento

Metaheurística	Sequencial	Paralelo
Evolução Diferencial	8 h e 25 min	2 h e 16 min
Algoritmo de Otimização de Leão	10 h e 47 min	3 h e 49 min
Busca Local Iterativa	11 h e 08 min	5 h e 31 min
ED com AOL	16 h e 19 min	6 h e 45 min
ED com BLI	10 h e 04 min	6 h e 52 min
AOL com BLI	10 h e 10 min	6 h e 44 min

Fonte: Autora (2018).

Da Tabela 16 é possível observar que a utilização da forma de processamento em paralelo traz uma diminuição, no pior caso, de aproximadamente 35% em relação ao processamento sequencial. Sendo que os parâmetros estimados tiveram seus respectivos valores alcançados e resultaram na saída aproximada ao serem comparados com a saída do sistema real obtido por dados de medições.

5.3 SISTEMA DE CONTROLE DE VELOCIDADE

O terceiro estudo de caso trata da estimação de parâmetros do sistema de controle de velocidade, sendo composto pela malha hidráulica e o conjunto conduto turbina do tipo Kaplan. Este sistema é um modelo caixa cinza onde os dados de entrada e saída, a estrutura do sistema e alguns dos seus parâmetros são conhecidos. A Figura 34 ilustra o sistema completo de uma usina hidrelétrica em diagrama de blocos.

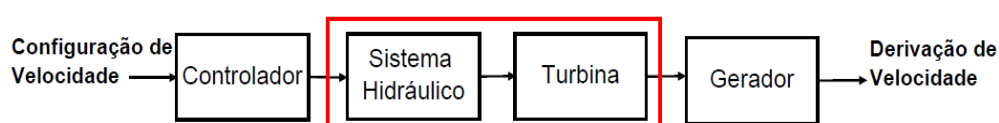


Figura 34 – Diagrama de blocos da usina Hidrelétrica.
Fonte: Adaptada de Kishor, Saini, Singh (2007).

Expandindo os blocos circundados em vermelho na Figura 34, tem-se que o sistema hidráulico envolve a válvula proporcional/distribuidora e distribuidor, e a turbina corresponde ao conjunto conduto turbina, conforme ilustrado na Figura 35.

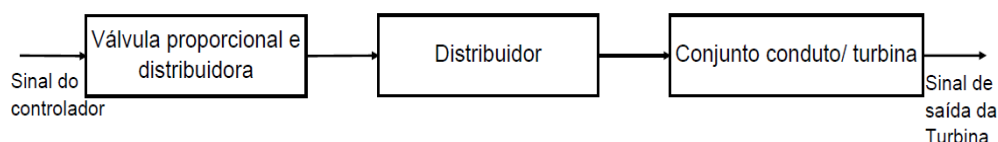


Figura 35 – Diagrama de blocos do sistema hidráulico com a turbina.
Fonte: Autora (2018).

A turbina Kaplan tem o mesmo funcionamento da Francis, mas suas pás são ajustáveis, com isto, se consegue realizar a inclinação delas proporcionando um melhor rendimento em toda sua faixa de funcionamento.

O modelo desenvolvido para o sistema identificado é mostrado na Figura 36, onde se tem como entrada a saída do controlador de velocidade e como saída a potência mecânica da turbina. Nesta figura são mostrados o modelo do sistema da malha hidráulica contendo a válvula proporcional/distribuidora, distribuidor e o conjunto conduto/turbina com seus respectivos parâmetros a ser estimados pelas técnicas de otimização.

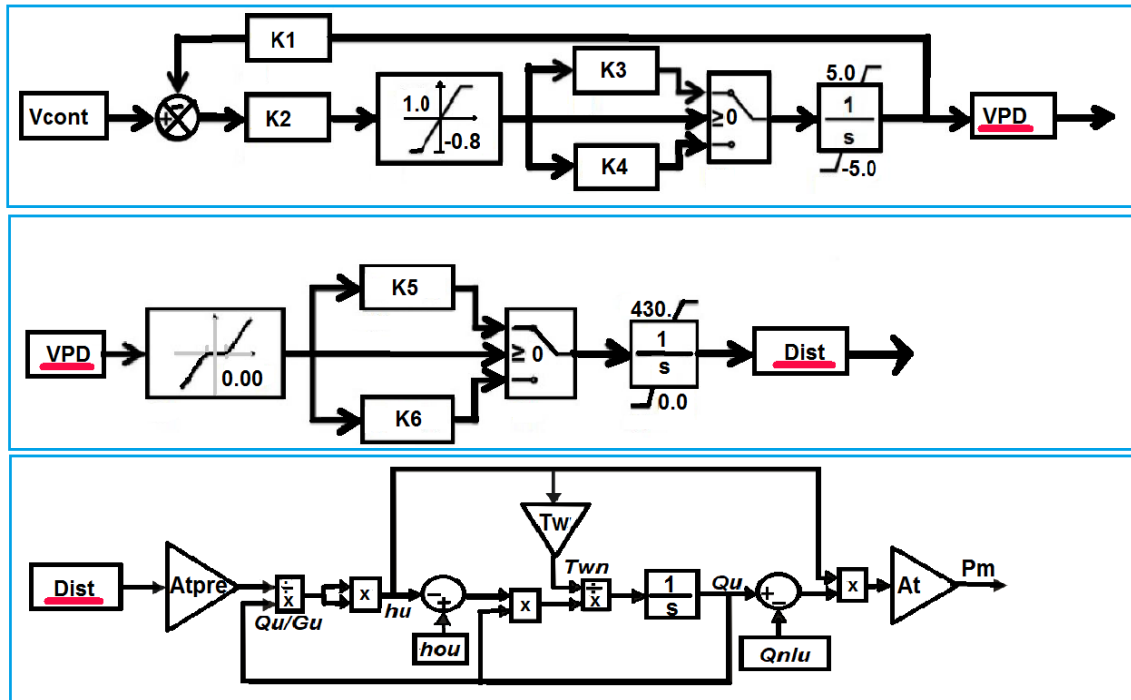


Figura 36 – Modelo matemático da malha hidráulica.
Fonte: Adaptada de Osinski (2017).

O modelo apresentado na Figura 36 funciona tanto em regime permanente quanto regime transitório e é o utilizado pela SIN em suas simulações.

A entrada do modelo são os dados obtidos do controlador (V_{cont}), a saída é representada pela válvula proporcional e distribuidora (VPD) em pu que será à entrada do distribuidor ($Dist$) em pu, $K1, K2, K3$ e $K4$ são os ganhos da válvula proporcional e distribuidora, $K5$ e $K6$, são ganhos do distribuidor, a saída $Dist$ é à entrada do conjunto conduto/turbina, a saída do conjunto conduto/turbina é representada pela potência mecânica (P_m) em pu, G_u é a posição do gate do distribuidor em pu, Q_u é a vazão em pu, h_u é a altura em pu, h_{ou} é o valor de estado estacionário inicial de h_u em pu, T_w é a constante de tempo de inércia da água em segundos onde não se tem seu valor, Q_{nlu} é a vazão necessária para suprir as perdas em vazio da turbina em pu, $1/s$ é o integrador, A_{tpre} e A_t são os respectivos ganhos do sistema.

Para realizar a estimação de parâmetros desse modelo não linear, a Tabela 17 ilustra as configurações principais das metaheurísticas utilizadas para este estudo de caso.

Tabela 17 – Configuração das metaheurísticas

Algoritmo	<i>estr_F</i>	<i>F</i>	<i>Cr</i>	<i>Id</i>	<i>tamBusca</i>	<i>P</i>
ED	DE/best/1	0,2 - 0,8	0,7	-	-	-
AOL	Binário	0,3	0,6	4	-	-
BLI	-	-	-	-	8	$Np/2$

Fonte: Autora (2018).

A taxa de mutação é definida por *F*, a taxa de cruzamento por *Cr*, a idade adulta dos leões é *Id*, o tamanho da busca local é *tamBusca* e a perturbação aplicada nas soluções ótimas é *P*.

Para os parâmetros estimados, algumas restrições para eles foram impostas. A Tabela 18 mostra essas restrições onde os limites inferior (*Bl*) e superior (*Bu*) são as restrições dos parâmetros.

Tabela 18 – Configuração dos parâmetros.

Parâmetros	Restrições
K1	0,5 a 0,8
K2	2 a 2,2
K3	2,5 a 2,9
K4	2 a 2,7
K5	14 a 26
K6	0,1 a 6,5
<i>T_w</i>	2 a 3
<i>A_t</i>	1,1 a 1,8

Fonte: Autora (2018).

Para o parâmetro A_{tpre} , o valor foi definido pela a equação (5.3):

$$A_{tpre} = \text{condição inicial} * 0,0023 * A_t \quad (5.3)$$

onde a condição inicial é igual a 267,5.

Como configuração, a população tem tamanho igual a 80 ($N = 80$), a dimensão é a quantidade parâmetros a ser estimados, que é igual a oito parâmetros ($dim = 8$); com duzentas e cinquenta iterações ($iter = 250$), critério de parada 10^{-3}

(*stopCriterio* = 0,001) e foram realizadas duzentas e cinquenta realizações (*count* = 250) para a obtenção dos valores mínimo, máximo, médio e desvio padrão (Σ) que são mostradas na Tabela 19.

Tabela 19 – Parâmetros estimados pelas metaheurísticas.

Metaheurística ED	Mín.	Máx.	Méd.	Σ
K1	0,5005	0,5999	0,5503	0,0292
K2	2,1602	2,1899	2,1751	0,0088
K3	2,6517	2,7993	2,7299	0,0433
K4	2,5005	2,6999	2,6002	0,0595
K5	21,0017	25,985	21,4783	0,2846
K6	3,3012	3,9972	3,6925	0,2103
A_{tpre}	0,4389	0,4562	0,4480	0,0051
T_w	2,8009	2,9993	2,9060	0,0584
A_t	1,6018	1,7989	1,7027	0,0566
Metaheurística AOL	Mín.	Máx.	Méd.	Σ
K1	0,5524	0,795	0,6658	0,0707
K2	1,90	1,997	1,9541	0,0278
K3	0,1463	6,5375	2,2794	1,8216
K4	2,0002	2,6936	2,3199	0,2218
K5	20,0761	25,963	22,985	1,7638
K6	0,1463	6,5375	3,2794	1,8216
A_{tpre}	0,4662	0,5427	0,5025	0,0209
T_w	2,1072	2,6982	2,4239	0,1674
A_t	1,5001	1,6994	1,5961	0,0575
Metaheurística BLI	Mín.	Máx.	Méd.	Σ
K1	0,5513	0,7994	0,6776	0,0771
K2	1,9001	1,9996	1,9511	0,0281
K3	2,0025	2,7931	2,3770	0,2374
K4	2,0100	2,6987	2,3401	0,1997
K5	20,0099	25,902	23,0694	1,6819
K6	0,1275	6,5970	3,3862	1,9732

A_{tpre}	0,4664	0,5432	0,5006	0,0218
T_w	2,1006	2,6912	2,3962	0,1817
A_t	1,5018	1,6989	1,6031	0,0548
Metaheurística ED com AOL	Mín.	Máx.	Méd.	Σ
K1	0,5510	0,7991	0,6811	0,0715
K2	1,9001	1,9999	1,9492	0,0295
K3	2,007	2,7969	2,3977	0,2360
K4	2,0048	2,7000	2,3694	0,2007
K5	20,0039	25,993	22,8419	1,7373
K6	0,1811	6,5993	3,5173	1,9226
A_{tpre}	0,4673	0,5432	0,4995	0,0217
T_w	2,1008	2,6860	2,4143	0,1685
A_t	1,5003	1,6994	1,6022	0,0609
Metaheurística ED com BLI	Mín.	Máx.	Méd.	Σ
K1	0,5510	0,7970	0,6728	0,0723
K2	1,9019	1,9995	1,9451	0,0285
K3	2,0216	2,7969	2,4429	0,2322
K4	2,0096	2,6948	2,3667	0,2180
K5	20,0450	25,981	23,0199	1,6111
K6	0,3586	6,5729	3,5640	1,7410
A_{tpre}	0,4673	0,5432	0,4997	0,0219
T_w	2,1008	2,6860	2,4125	0,1695
A_t	1,5007	1,6994	1,6027	0,0619
Metaheurística AOL com BLI	Mín.	Máx.	Méd.	Σ
K1	0,5514	0,7998	0,6786	0,0727
K2	1,9001	1,9999	1,9499	0,0296
K3	2,0010	2,7981	2,3382	0,2429
K4	2,0030	2,6989	2,3570	0,2030
K5	20,0188	25,991	23,0694	1,7238
K6	0,1406	6,5578	3,3613	1,8670
A_{tpre}	0,4662	0,5433	0,4981	0,0213

T_w	2,1002	2,6969	2,4252	0,1672
A_t	1,5016	1,6994	1,6087	0,0591

Fonte: Autora (2018).

Com os resultados da Tabela 19, é possível ilustrar graficamente a variação dos parâmetros pelos algoritmos individuais e colaborativos.

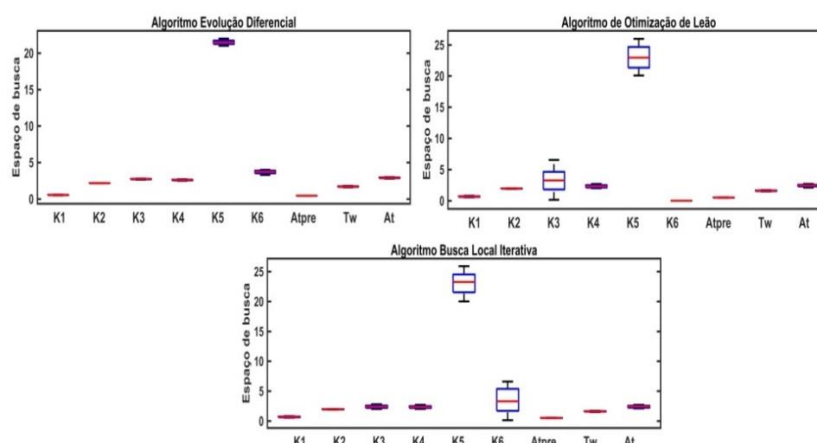


Figura 37 – Variação dos parâmetros das metaheurísticas individuais.
Fonte: Autora (2018).

A variância dos parâmetros estimados pelos algoritmos individuais, mostrados na Figura 37, ocorre maior variância nos parâmetros $K3$ e $K5$ do algoritmo AOL e $K5$ e $K6$ do algoritmo BLI. Já os parâmetros estimados pelo algoritmo ED mostraram menor variância ao comparar com os algoritmos AOL E BLI.

Já os algoritmos colaborativos, apresentam variância maior de pelo menos dois parâmetros estimados por cada técnica colaborativa. Para as técnicas, todas elas apresentaram maior variância nos parâmetros $K2$ e $K5$, e obteve melhora da variância para os parâmetros $K6$.

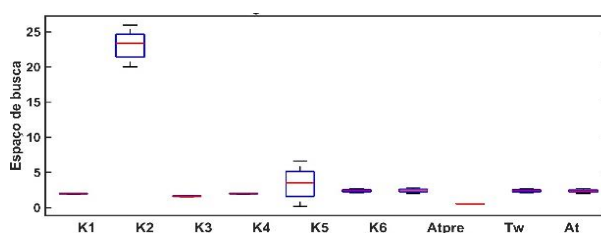


Figura 38 – Variância da técnica colaborativa AOL com BLI.
Fonte: Autora (2018).

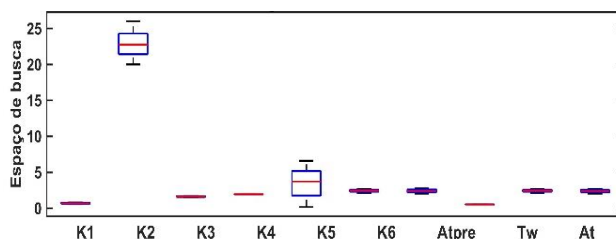


Figura 39 – Variância da técnica colaborativa ED com AOL.
Fonte: Autora (2018).

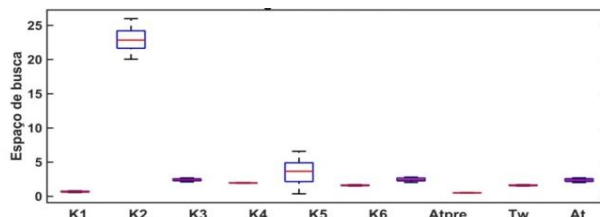


Figura 40 – Variância da técnica colaborativa ED com BLI.
Fonte: Autora (2018).

Com os valores da média de cada parâmetro foi possível encontrar a saída mais próxima do modelo ao ser comparada com os dados de campo.

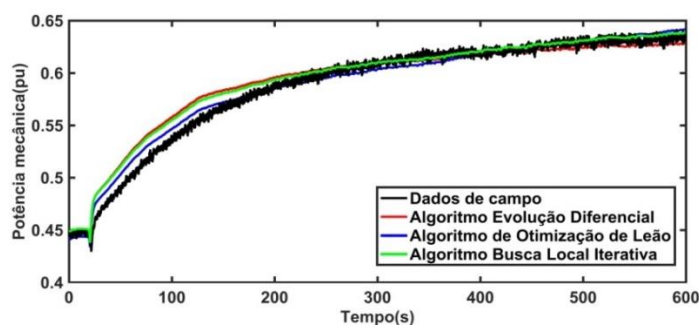


Figura 41 – Saída da malha hidráulica com as metaheurísticas individuais.
Fonte: Autora (2018).

Com a Figura 41, pode se perceber a aproximação da onda obtida metaheurísticas individuais com a onda obtida por dados de medições. Neste estudo de caso, o algoritmo AOL mostra ser mais eficiente do os outros algoritmos, pois foi o que conseguiu uma aproximação melhor. Já para a saída da malha hidráulica obtida pelas metaheurísticas colaborativas, a Figura 42 mostra a estabilização das curvas por cada técnica colaborativa.

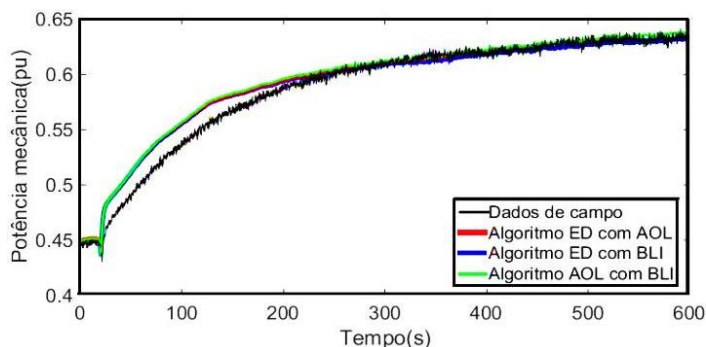


Figura 42 – Saída da malha hidráulica com metaheurísticas colaborativas.
Fonte: Autora (2018).

Observando as Figuras 41 e 42, pode-se concluir que o algoritmo AOL individual foi o melhor de todos. Neste estudo de caso, as técnicas colaborativas não consegue melhorar a saídas obtidas pelos algoritmos, mas também não ocorre degradação da saída. O fato de não encontrar a saída mais próxima da saída obtida por dados de medições envolve dois fatores, a filtragem dos dados que não excluiu todos os ruídos e a estimação dos parâmetros que com a melhora da variável $K6$, acabou implicando na variável $K2$ que não permitiu melhorar a saída do modelo identificado.

Além disso, foi realizada uma análise onde a diferença da saída referenciada e saída obtida através dos parâmetros estimados resultam no erro médio quadrático (EMQ).

Com isto, o valor mínimo, máximo, media e desvio padrão de cada algoritmo durante as realizações ($count = 500$) é mostrado na Tabela 20.

Tabela 20 – Variância do EMQ dos algoritmos individuais.

Comparação	ED	AOL	BLI
Mínimo	61,0339	23,2137	23,5549
Máximo	80,6125	83,8225	81,9163
Média	69,3958	50,1091	48,0714
Desvio padrão (Σ)	4,9509	14,4938	14,0084

Fonte: Autora (2018).

Com os resultados da Tabela 20, o algoritmo ED mostra uma variância menor ao ser comparado com os demais algoritmos individuais. Já os algoritmos AOL e BLI apresentam variâncias bem próximas uma da outra e conseguem o valor

médio melhor do que o algoritmo ED. Para os algoritmos colaborativos, a Tabela 21 ilustra os resultados obtidos.

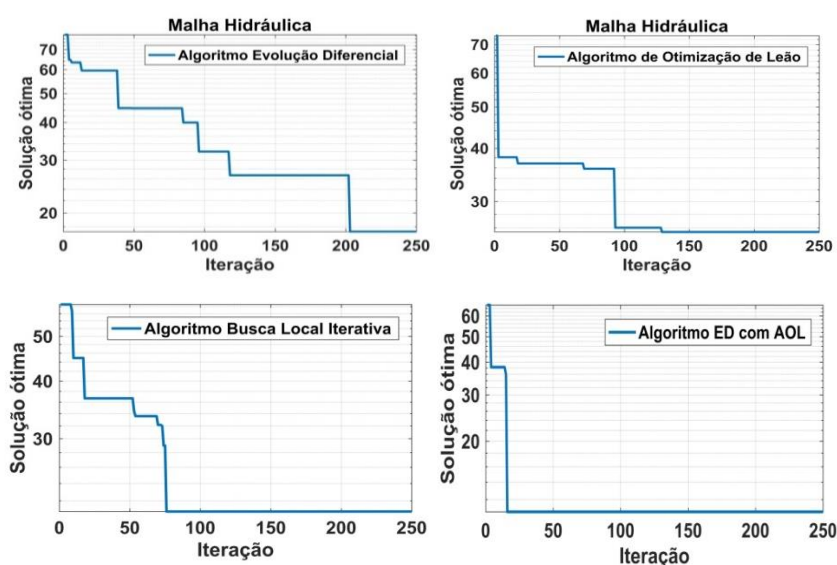
Tabela 21 – Variância do EMQ dos algoritmos individuais.

Comparação	ED com AOL	ED com BLI	AOL com BLI
Mínimo	19,5357	19,1557	20,4925
Máximo	82,3974	81,9163	85,8752
Média	47,2092	49,2725	48,9459
Desvio padrão (Σ)	14,6737	14,3880	14,8849

Fonte: Autora (2018).

Os algoritmos colaborativos mostram uma variância bem aproximada uma das outras, mas pode se dizer que o algoritmo ED com BLI alcançou a solução factível com menor variância.

Em relação à convergência de cada algoritmo durante todo o processo de otimização, pode-se dizer que todos os algoritmos encontraram boas soluções, mesmo os algoritmos colaborativos apresentando uma variância maior do que o algoritmo ED individual. Em relação à convergência de cada algoritmo, pode se observar que os algoritmos colaborativos convergem mais rápido. A Figura 43 ilustra a variação das soluções ótimas nas iterações.



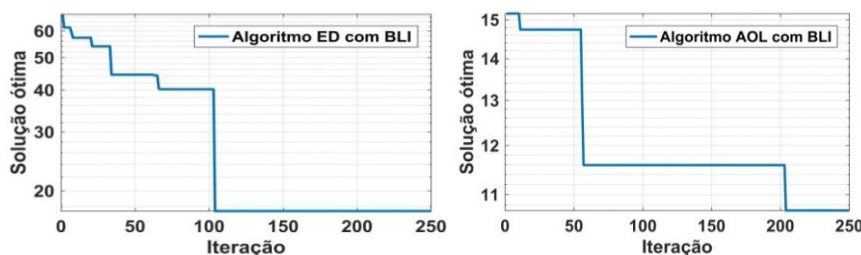


Figura 43 – Variação da solução ótima nas iterações.

Fonte: Autora (2018).

As soluções encontradas durante as iterações percebe-se que o algoritmo colaborativo ED com AOL converge mais rápido, antes de 50, do que os outros algoritmos individuais e colaborativos. Para realizar os testes, a população inicial dos algoritmos na primeira execução é igual para todos.

Para uma análise mais justa entre as metaheurísticas as Figuras 44 e 45 apresentam a variação do EMQ das respectivas metaheurísticas.

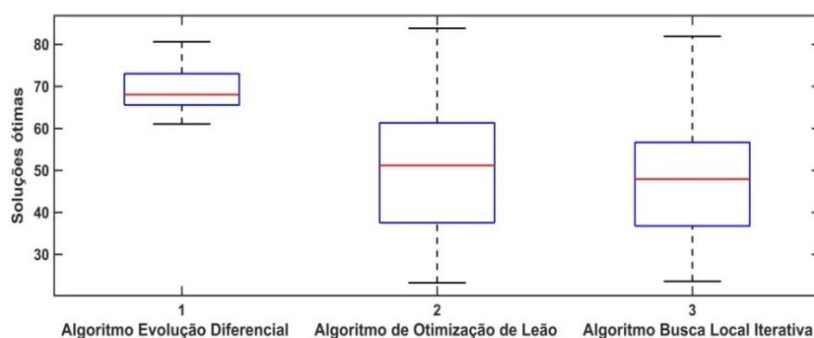


Figura 44 – EMQ das metaheurísticas individuais.

Fonte: Autora (2018).

A variância do EMQ obtida pelos algoritmos colaborativos está próximos uns dos outros, mas o algoritmo ED continua sendo o melhor para este estudo de caso. Mas quando se trata do valor médio da solução, fica visível que os algoritmos colaborativos resultam em valores abaixo de 50, como ilustra a Figura 45.

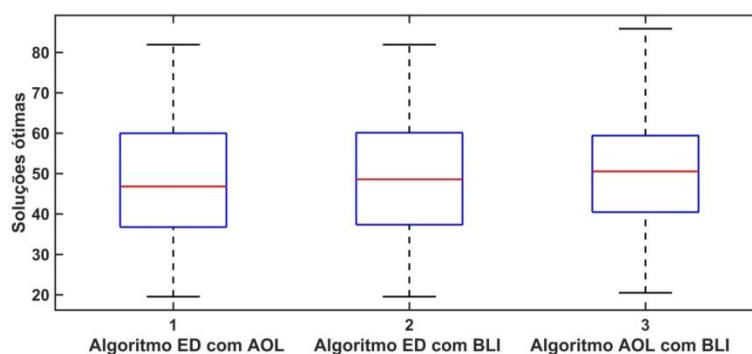


Figura 45 – EMQ das metaheurísticas colaborativas.
Fonte: Autora (2018).

Este estudo de caso não possível utilizar o paralelismo nos algoritmos, devido ao software Anatem® impossibilitar o uso do processamento paralelo. Em relação ao tempo de execução deste estudo de caso, a Tabela 22 ilustra os respectivos tempos de cada algoritmo.

Tabela 22 - Tempo de processamento

Metaheurística	Sequencial
Evolução Diferencial	18 h e 31 min
Algoritmo de Otimização de Leão	16 h e 06 min
Busca Local Iterativa	17 h e 58 min
ED com AOL	17 h e 20 min
ED com BLI	16 h e 35 min
AOL com BLI	16 h e 19 min

Fonte: Autora (2018).

Os parâmetros estimados tiveram seus respectivos valores alcançados e resultaram na saída aproximada ao serem comparados com a saída do sistema real obtido por dados de medições e observações. Além disso, o tempo de processamento é grande devido aos dados de campos e a quantidade de parâmetros estimados sobrecarrega o algoritmo ocorrendo lentidão no processo de busca por melhores soluções.

6 CONSIDERAÇÕES FINAIS

Neste trabalho foram exploradas as metaheurísticas Evolução Diferencial (ED), Algoritmo de Otimização de Leão (AOL) e Busca Local Iterativa (BLI) de modo colaborativo para tratar do problema da estimação de parâmetros de sistemas reais. Além disto, as metaheurísticas foram paralelizadas com o objetivo de diminuir o tempo computacional ao realizar a estimação das variáveis pelas metaheurísticas. Os dados dos sistemas reais utilizados neste trabalho são pertencentes ao SIN. Destes dados, foram realizados os estudos de casos dos sistemas: conjunto conduto/turbina do tipo Francis; e controlador de velocidade com o conjunto conduto/turbina do tipo Kaplan.

Inicialmente as metaheurísticas ED, AOL, BLI na forma sequencial e na forma colaborativa foram utilizadas em funções testes para verificar a acurácia de cada uma. Constatada a eficiência das metaheurísticas, estas foram então utilizadas em dois problemas reais.

Verificou-se que as metaheurísticas foram eficientes e confiáveis para lidar com o problema de identificação de sistemas, isto é, elas obtiveram sucesso ao estimar os parâmetros com valores que levaram a quase completa superposição das saídas do modelo com os dados de campo.

Em vista disto, a proposta das metaheurísticas para trabalhar na versão colaborativa, mostrou-se uma alternativa bastante atraente. Pois ao trocar informações pode se perceber que em alguns estudos de casos, como no conjunto conduto/turbina do tipo Francis, apresentam resultados melhores ao ser comparadas com as técnicas individuais. Além disto, nas duas funções testes de otimização, Ackely e Rosenbrock, apresentarão resultados significativos, podendo avaliar os algoritmos colaborativos como eficientes e prontos a serem aplicados em problemas reais do nosso cotidiano.

Outra contribuição deste trabalho, foi a implementação dos algoritmos em paralelo, com isto o tempo computacional de cada metaheurística foi analisado individualmente na forma sequencial e colaborativa. Quando a computação paralela é aplicada na função objetivo, ela acelera os cálculos da função durante todo o procedimento dos algoritmos. Ao analisar o tempo computacional de cada algoritmo, percebe se que a lentidão se dá em função da quantidade de variáveis encontradas e a definição do tamanho da população. Já em sistemas reais, a lentidão encontra

se nestes dois aspectos apresentados e na quantidade de dados de entrada do sistema identificado. Dessa maneira, o paralelismo aplicado nos algoritmos foi eficiente por conseguir diminuir o tempo computacional do estudo de caso conjunto conduto/turbina do tipo Francis em torno de 35% comparado quando os algoritmos processam de modo sequencial.

Para trabalhos futuros, o tempo computacional é um atrativo para as metaheurísticas, uma vez que são paralelizadas conseguem acelerar os processos de cada uma delas e obtendo boas soluções ótimas. Com isto, a identificação de sistemas dinâmicos requer técnicas de otimização cada vez mais robustas para solucionar problemas do dia a dia aplicando estas três metaheurísticas juntas de forma colaborativa com outros tipos de técnicas de otimização robustas.

REFERÊNCIAS

- AGUIRRE, L. A. Introdução à Identificação de Sistemas – Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais. **UFMG**: Belo Horizonte. 3ª edição, p.730, 2007.
- AGUIRRE, L. A.; BRUCIAPAGLIA, A. H.; MIYAGI, P. E.; PIQUEIRA, J. R. C. **Enciclopédia de automática : controle e automação**. São Paulo: Blucher, 2007.
- ALBA, E. **Parallel a New Class of Algorithms**. 1ª ed. New Jersey, USA, 2005.
- ANITA CHRISTALINE, J.; RAMESH, R.; VAISHALI, D. Bio-inspired computational algorithms for improved image steganalysis. **Indian Journal of Science and Technology**, v. 9, 2016.
- AO, Y.; CHI, H. Experimental Study on Differential Evolution Strategies. **2009 WRI Global Congress on Intelligent Systems**, p. 19–24, 2009.
- ATYABI, A.; SAMADZADEGAN, S. Particle Swarm Optimization : A Survey. **ResearchGate**, p. 1354–1361, 2011.
- BÄCK, T.; FOGEL, D. B.; MICHALEWICZ, Z. Handbook of Evolutionary Computation. **Evolutionary Computation**, v. 2, p. 1–11, 1997.
- BARROSO, M. F. S.; TAKAHASHI, R. H. C.; AGUIRRE, L. A. Multi-objective parameter estimation via minimal correlation criterion. **Journal of Process Control**, v. 17, n. 4, p. 321–332, 2007.
- BERGAMINI, M. G.; OLIVEIRA, J. L.; OLIVEIRA, G. H. C.; LEANDRO, G. V. Estimação de Parâmetros de Sistemas Não Lineares Utilizando Metaheurísticas Híbridas Paralelizadas. **XIII Brazilian Congress on Computational Intelligence**, p. 11, 2017.
- BERTACHI, A. H. Otimização de parâmetros via metaheurísticas populacionais e validação de um controlador de estrutura variável. , 2014.
- BIRATTARI, M.; PAQUETE, L.; STÜTZLE, T.; VARRENTTRAPP, K. Classification os Metaheuristics and Design of Experiments for the Analysis of Components. **Intellectics Group**, p. 1–12, 2001. Darmstadt.
- BORTOFF, S.; SPONG, M. Parameter identification for nonlinear systems. **Proceedings of the 29th Conference on Decision and Control , Honolulu,Hawaii**, p. 772–777, 1990.
- BOSCO, M. C.; GUEDES, J. J.; CASTOLDI, M. F.; et al. Estimation of Parameters and Tuning of a Speed PI of Permanent Magnet DC Motor Using Differential Evolution. **Electric Machines and Drives Conference (IEMDC)**, 2017.
- BOUSSAÏD, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. **Information Sciences**, v. 237, p. 82–117, 2013.
- ÇANAKOĞLU, A. I.; YETGIN, A. G.; TEMURTAŞ, H.; TURAN, M. Induction Motor

Parameter Estimation Using Metaheuristic Methods. **Journal of Electrical Engineering and Computer Sciences**, p. 1177–1192, 2014.

CHU, H.-J.; CHANG, L.-C. Applying Particle Swarm Optimization to Parameter Estimation of the Nonlinear Muskingum Model. **Journal of Hydrologic Engineering**, v. 14, n. 9, p. 1024–1027, 2009.

CLERC, M. **Particle Swarm Optimization**. John Wiley & Sons, 2010.

COELHO, A. A. R.; COELHO, L. D. S. **Identificacao de Sistemas Lineares Dinâmicos**. Florianópolis, SC, 2004.

COELHO, L. DOS S. Notas em Matemática Aplicada. **Sociedade Brasileira de Matemática Aplicada e Computacional**, v. 2, p. 99, 2003.

CORRÊA, M. V.; AGUIRRE, L. A.; SALDANHA, R. R. Using steady-state prior knowledge to constrain parameter estimates in nonlinear system identification. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, p. 1376–1381, 2002.

CORRÊA, M. V.; AGUIRRE, L. Identificação não-linear caixa-cinza: Uma revisão e novos resultados. **Revista Controle & Automação**, v. 15, p. 18, 2004.

COTTA, C.; TALBI, E. G.; ALBA, E. Parallel Hybrid Metaheuristics. **ResearchGate**, p. 347–370, 2005.

CRAINIC, T. G.; TOULOUSE, M. Parallel strategies for meta-heuristics. **Handbook of Metaheuristics**, p. 475–513, 2003.

DAS, R. Parameter Estimation of a Space Radiator using Differential Evolution Algorithm. **Ninth International Conference on Contemporary Computing (IC3)**, p. 6, 2016.

DIBBLEE, D.; MALTESE, J.; OMBUKI-BERMAN, B. M.; ENGELBRECHT, A. P. Vector-Evaluated Particle Swarm Optimization With Local Search. **2015 IEEE Congress on Evolutionary Computation (CEC'2015)**, p. 187–195, 2015.

DRÉO, J.; AUMASSON, J.-P.; TFAILI, W.; SIARRY, P. Adaptive Learning Search, a New Tool To Help Comprehending Metaheuristics. **International Journal on Artificial Intelligence Tools**, v. 16, p. 1–23, 2007.

EIBEN, A.E. ; SMITH, J. E. **Introduction to Evolutionary Computing**. 2^o Edição ed. New York, 2004.

ESPOSITO, A.; GOMES, H. M. COMPUTAÇÃO PARALELA E SERIAL APLICADAS À OTIMIZAÇÃO. **Asociación Argentina de Mecánica Computacional**, v. XXXI, p. 925–943, 2012.

FREITAS, U. S.; MACAU, E. E. N. Estimção de Parâmetros em Sistemas Não Lineares Usando Controle Geométrico. **WORKSHOP DOS CURSOS DE COMPUTAÇÃO APLICADA DO INPE**, 2004. São José dos Campos: INPE, 2004. CD-ROM, On-line.

GENDREAU, M.; POTVIN, J. **Handbook of Metaheuristics**. 2^a ed. New York, NY,

USA: Springer, 2010.

GOMES, M. C.; LEANDRO, G. V.; RIBEIRO, E. P. Estimação de Parâmetros de Sistemas Não Lineares Utilizando Algoritmos Genéticos. **Anais do IX Congresso Brasileiro de Redes Neurais/ Inteligência COmputacional (IX CBRN)**, p. 8, 2009.

GUI, T.; MA, C.; WANG, F.; WILKINS, D. E. Survey on swarm intelligence based routing protocols for wireless sensor networks: An extensive study. **IEEE International Conference on Industrial Technology (ICIT)**, p. 1944–1949, 2016. Taipei -Taiwan.

IEEE Working Group on Prime Mover and Energy Supply Models for System Dynamic Performance Studies, “Hydraulic Turbine and Turbine Control Models for System Dynamic Studies,” in **IEEE Transactions on Power Systems**, v. 7, n. 1, pp. 167–179, Feb. 1992.

ISERMANN, R.; MÜNCHHOF, M. **Identification of Dynamic Systems**. New York, 2011.

JOUNI A. LAMPINEN, KENNETH V. PRICE, R. M. S. **Differential Evolution - A Practical Approach to Global Optimization**. 2008.

KISHOR, N.; SAINI, R. P.; SINGH, S. P. A review on hydropower plant models and control. **Renewable and Sustainable Energy Reviews**, v. 11, n. 5, p. 776–796, 2007.

KUSHIDA, J. I.; HARA, A.; TAKAHAMA, T. Rank-based differential evolution with multiple mutation strategies for large scale global optimization. **2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings**, , n. 1, p. 353–360, 2015.

LIPOWSKI, A.; LIPOWSKA, D. Roulette-wheel selection via stochastic acceptance. **Physica A: Statistical Mechanics and its Applications**, v. 391, n. 6, p. 2193–2196, 2012. Elsevier B.V.

LJUNG, L. **System Identification Theory for User**. New Jersey, 1987.

LJUNG, L. Perspectives on System Identification. **Annual Reviews in Control**, v. 34, p. 12, 2010.

LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. **ITERATED LOCAL SEARCH**. New York: Springer US, 2003.

MARGOTI, L. M.; SANTOS, A. P. L.; MILAGRES, N. R. L.; et al. Aplicação de Representações em Blocos Interconectados em Identificação Caixa-cinza de Sistemas Dinâmicos Não-lineares. **XVIII Congresso Brasileiro de Automática**, p. 4224–4230, 2010.

MELO, F. P. DE; KOESSLER, R. J. Hydraulic Turbine and Turbine Control Models for System Dynamic Studies. **Transactions on Power Systems**, v. 7, p. 167–179, 1992.

MELO, W. A. X.; FAMPA, M. H. C.; RAUPP, F. M. P. BUSCA LOCAL INTENSIVA: UMA NOVA METAHEURÍSTICA PARA OTIMIZAÇÃO GLOBAL RESTRITA. **Anais do XLI Simpósio Brasileiro de Pesquisa Operacional, SBPO**, 2009.

MOHD AZMI, K. Z.; PEBRIANTI, D.; IBRAHIM, Z.; SUDIN, S.; NAWAWI, S. W. Simultaneous computation of model order and parameter estimation for system identification based on gravitational search algorithm. **The 6th International Conference on Intelligent Systems, Modelling and Simulation**, , n. 1, p. 135–140, 2015.

MOSSER, A. A. **Group territoriality of the African lion: behavioral adaptation in a heterogeneous landscape**. Minnesota, EUA, 2008.

MULATI, M. H.; CONSTANTINO, A. A.; SILVA, A. F. DA. **Otimização por Colônia de Formigas**. 1ª ed. Curitiba - PR, 2013.

NAVARRO, C. M.; CARVALHO, A. S.; JUNIOR, G. G. P. ESTIMAÇÃO DE PARÂMETROS DE UM MOTOR CC POR MEIO DE REDES NEURAS ARTIFICIAIS (rna). **XLII SBPO**, p. 1480–1490, 2009.

NELLES, O. **Nonlinear System Identification: from Classical Approaches to Neural Networks and Fuzzy Models**. Springer-V ed. Berlin, 2001.

OSINSKI, C. **MODELAGEM DE MALHA HIDRÁULICA E CONTROLE DE VELOCIDADE DE TURBINAS HIDRÁULICAS EM UMA USINA HIDRELÉTRICA**, 2017. Universidade Federal do Paraná.

PANT, M.; ALI, M.; ABRAHAM, A. Mixed mutation strategy embedded differential evolution. **2009 IEEE Congress on Evolutionary Computation, CEC 2009**, p. 1240–1246, 2009.

PEARSON, R. K.; POTTMANN, M. Gray-box identification of block-oriented nonlinear models. **Journal of Process Control**, v. 10, n. 4, p. 301–315, 2000.

PRATA, D. M.; SCHWAAB, M.; LIMA, E. L.; PINTO, J. C. Nonlinear dynamic data reconciliation and parameter estimation through particle swarm optimization: Application for an industrial polypropylene reactor. **Chemical Engineering Science**, v. 64, n. 18, p. 3953–3967, 2009.

RACHAD, S.; FOURAIJI, H.; BENSASSI, B. Modeling a production system by parametric identification approach. **World Conference on Complex Systems**, p. 402–406, 2015.

RAJAKUMAR, B. R. The Lion's Algorithm: A New Nature-Inspired Search Algorithm. **Procedia Technology**, v. 6, p. 126–135, 2012. B.R. Rajakumar.

RAJAKUMAR, B. R. Lion algorithm for standard and large scale bilinear system identification: A global optimization based on Lion's social behavior. **IEEE Congress on Evolutionary Computation**, v. 2, p. 2116–2123, 2014.

RANNEN, S.; GHORBEL, C. Identification of Complex Systems Using Nonlinear Discrete Polynomial Structures. **Internacional Conference on Advanced Systems and Electric Technologies**, p. 38–43, 2017.

RAO, S. S. **Engineering Optimization: Theory and Practice**. 4ª ed. New Jersey, USA: JOHN WILEY & SONS, INC, 2009.

SAIT, S. M.; KHAN, K. S.; ALI, M. I. Parallel strategies for stochastic evolution.

Proceedings of The 7th International Conference on Intelligent Systems Design and Applications, ISDA 2007, p. 813–818, 2007.

SALCEDO-SANZ, S. Modern meta-heuristics based on nonlinear physics processes: A review of models and design procedures. **Physics Reports**, v. 655, p. 1–70, 2016. Elsevier B.V.

SCHWAAB, M.; BISCAIA, E. C.; MONTEIRO, J. L.; PINTO, J. C. Nonlinear parameter estimation through particle swarm optimization. **Chemical Engineering Science**, p. 1542–1552, 2008.

SEVERINO, A. G. V.; ARAÚJO, Í. B. Q.; LINHARES, L. L. DA S.; ARAÚJO, F. M. U. Metaheurísticas para Estimação de Parâmetros na Identificação de Sistema Não Lineares Utilizando Modelos NARMAX Polinomiais. **IX Congresso Nacional de Engenharia Mecânica**, p. 8, 2016.

STORN, R.; PRICE, K. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. **Journal of Global Optimization**, v. 11, n. 4, p. 341–359, 1997.

SUN, J.; GARIBALDI, J. M.; HODGMAN, C. Parameter estimation using metaheuristics in systems biology: A comprehensive review. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, v. 9, p. 185–202, 2012.

TALBI, E. G. **Metaheuristics: From Design to Implementation**. Canada, 2009.

TANGHERLONI, A.; NOBILE, M. S.; CAZZANIGA, P. GPU-powered Bat Algorithm for the parameter estimation of biochemical kinetic values. **CIBCB 2016 - Annual IEEE International Conference on Computational Intelligence in Bioinformatics and Computational Biology**, 2016.

VIANNA, D.; VIANNA, M. Local search-based heuristics for the multiobjective multidimensional knapsack problem. **Produção**, p. 478–487, 2013.

WANG, B.; JIN, X. P.; CHENG, B. Lion pride optimizer: An optimization algorithm inspired by lion pride behavior. **Science China Information Sciences**, v. 55, n. 10, p. 2369–2389, 2012.

YANG, J. F.; LIN, Y. DA; CHUANG, L. Y.; YANG, C. H. An improved GA for identifying susceptibility genes in the presence of epistasis. **2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings**, p. 1–8, 2015.

YAZDANI, M.; JOLAI, F. Lion Optimization Algorithm (LOA): A nature-inspired metaheuristic algorithm. **Journal of Computational Design and Engineering**, v. 3, n. 1, p. 24–36, 2016. Elsevier.